

RESEARCH NOTE 80-15

**LEVEL II**



AD A092208

**PROGRAMMER'S MANUAL FOR THE  
TACTICAL OPERATIONS SYSTEM  
ANALYSIS PACKAGE**

Gary Witus, Mark Meerschaert, Donald Kleist

VECTOR RESEARCH, INCORPORATED  
P.O. Box 1506  
Ann Arbor, Michigan 48106

Human Factors Technical Area



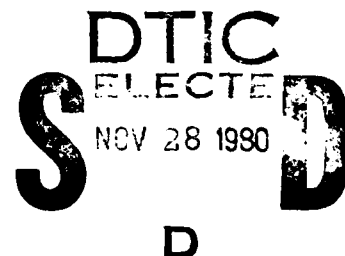
U. S. Army

Research Institute for the Behavioral and Social Sciences

May 1980

Contract DAHC19-78-C-0027

Approved for public release; distribution unlimited.



DDC FILE COPY

80 11 25 043

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM															
1. REPORT NUMBER (18) ARI Research Notes 80-15/	2. GOVT ACCESSION NO. AD-A092 208	3. RECIPIENT'S CATALOG NUMBER (19) RN-1															
4. TITLE (and Subtitle) (6) Programmer's Manual for the Tactical Operations System Analysis Package.		5. TYPE OF REPORT & PERIOD COVERED (7) Interim May 79-May 80															
7. AUTHOR(s) (10) Gary/Witus, Mark/Meerschaert, Donald/Kleist		6. PERFORMING ORG. REPORT NUMBER (14) VRI-ARI-3-FR80-4															
9. PERFORMING ORGANIZATION NAME AND ADDRESS Vector Research, Incorporated, PO Box 1506 Ann Arbor, Michigan 48106		8. CONTRACT OR GRANT NUMBER(s) (15) DAHC19-78-C-0027															
11. CONTROLLING OFFICE NAME AND ADDRESS CORADCOM Field Office ATTN: DRDCO-FL Ft. Leavenworth, Kansas 66027		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) 2Q363739A793 (12) 1A Task 1A Workunit 001															
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) US Army Research Institute for the Behavioral and Social Sciences ATTN: PERI-OS 5001 Eisenhower Ave Alexandria VA 22333		12. REPORT DATE 25 February 1980															
		13. NUMBER OF PAGES 175															
		15. SECURITY CLASS. (of this report) Unclassified															
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE															
16. DISTRIBUTION STATEMENT (of this Report)  Approved for open release; distribution unlimited.																	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)																	
18. SUPPLEMENTARY NOTES  This research was technically monitored by Mr. Robert S. Andrews and Dr. Stanley M. Halpin of ARI and LTC Larry Walker of CORADCOM.																	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) <table border="0"> <tr> <td>Command and Control</td> <td>Information Management</td> <td>Standing Operating</td> </tr> <tr> <td>Communications</td> <td>Information Summarization</td> <td>Procedures (SOP)</td> </tr> <tr> <td>Computers</td> <td>Mathematical Modeling</td> <td>Tactical Operations</td> </tr> <tr> <td>Data base Management</td> <td>Networks</td> <td>System (TOS)</td> </tr> <tr> <td>Design aid</td> <td>Queueing</td> <td></td> </tr> </table>			Command and Control	Information Management	Standing Operating	Communications	Information Summarization	Procedures (SOP)	Computers	Mathematical Modeling	Tactical Operations	Data base Management	Networks	System (TOS)	Design aid	Queueing	
Command and Control	Information Management	Standing Operating															
Communications	Information Summarization	Procedures (SOP)															
Computers	Mathematical Modeling	Tactical Operations															
Data base Management	Networks	System (TOS)															
Design aid	Queueing																
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) <p>This document discusses the programming details of a computer program package for the analysis of the Tactical Operations System (TOS). The program computer code is included. These programs were produced during the second phase of a project to develop information management concepts and procedures for automated battlefield command and control systems.</p>																	

## PREFACE

This document is one of eight reports which describe the work performed by Vector Research, Incorporated (VRI) and its subcontractor, Perceptronics, Incorporated, for the US Army Research Institute for the Behavioral and Social Sciences (ARI) under the second phase of contract number DAHC19-78-C-0027. The work described was performed over 12 months of an anticipated 36-month three-phased project. The overall objective of the project has been to produce procedural guidelines to be used by divisions in the field in developing standard operating procedures for information management in the Tactical Operations System (TOS). As a consequence of the redirection of the TOS development effort in November 1979, the objective of this work was reinterpreted to include automated battlefield command control systems (ABCCS) in general, using TOS for an explicit example of the design, human factors, and management control considerations which must be addressed.

The VRI study team for phase II was comprised of Dr. Robert W. Blum (Project Leader), Ms. Cathleen A. Callahan, Dr. W. Peter Cherry, Mr. Mark G. Graulich, Mr. Donald Kleist, Mr. Mark Meerschaert, Mr. Gregory Touma, and Mr. Gary Witus. The Perceptronics team for phase II consisted of Dr. Michael G. Samet and Dr. Ralph E. Geiselman.

The authors wish to acknowledge the helpful contributions of Dr. Stanley M. Halpin and Mr. Robert Andrews, who were charged with monitoring the study for ARI; and LTC L. Walker, MAJ. A. Edmonds, and Mr. M. Carrio, who performed a similar function for that portion of the study effort which was jointly sponsored with ARI by the US Army Communications Research and Development Command (CORADCOM).

The eight reports are as follows:

Blum et al. Information Management for an Automated Battlefield Command and Control System: Executive Summary. ARI Research Report 1249. -- presents an overview of the project and the other seven reports.

Callahan et al. Guidelines for Managing the Flow of Information in an Automated Battlefield Command and Control System. ARI Research Report 1348. -- describes considerations in and procedures for the management of contemporary ABCC systems.

Geiselman and Samet. Guideline Development for Summarization of Tactical Data. ARI Technical Report 458. -- an analysis of procedures for the extraction, summarization, and presentation of critical information.

Witus et al. Analysis of Information Flow in the Tactical Operations System (TOS). ARI Research Notes 80-12. -- describes the purpose, approach, and results of a TOS analysis which focused on TOS when integrated with a planned communications support system.

Witus et al. Description of the Tactical Operations System Information Flow Model. ARI Research Notes 80-13. -- describes the representation of TOS used to develop the analysis package and the mathematics of the model.

Witus et al. User's Manual for the Tactical Operations System Analysis Package. ARI Research Notes 80-14. -- explains the use and operation of the analysis package.

Witus et al. Programmer's Manual for the Tactical Operations System Analysis Package. ARI Research Notes 80-15. -- describes the programming details of the package to facilitate modifications or transfer between host systems.

Cherry, W. All Source Analysis System: Design Issues. ARI Working Paper HF80-XX. -- a discussion of design issues associated with the emerging ASAS concept.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

# CONTENTS

	<u>Page</u>
1.0 INTRODUCTION . . . . .	1-1
2.0 PROGRAM CREATE . . . . .	2-1
2.1 Variables in Program Create . . . . .	2-1
2.2 Basic Flow of Program Create . . . . .	2-13
2.3 Subroutines in Create . . . . .	2-17
2.4 Data File . . . . .	2-19
2.5 Code of Program Create . . . . .	2-22
3.0 PROGRAM MODIFY . . . . .	3-1
3.1 Structure of Program MODIFY . . . . .	3-1
3.2 Code of Program Modify . . . . .	3-3
3.3 Flow Charts of the Program Modify and Selected Subroutines . . . . .	3-5
4.0 PROGRAM DISPLAY . . . . .	4-1
5.0 PROGRAM COMPUTE . . . . .	5-1
APPENDIX A . . . . .	A-1
APPENDIX B . . . . .	B-1
APPENDIX C . . . . .	C-1

## EXHIBITS

<u>Number</u>	<u>Title</u>	<u>Page</u>
1-1	Analysis Package Organization	1-2
2-1	Configuration Variables in CREATE	2-2
2-2	Communications System Variables in CREATE	2-5
2-3	Message Variables in CREATE	2-6
2-4	Route Cross Message Array Variables in CREATE	2-7
2-5	Engineering Data in CREATE	2-8
2-6	Environmental Variables in CREATE	2-11
2-7	Logic Variables in CREATE	2-12
2-8	Basic Flow of CREATE	2-14
2-9	Subroutines in CREATE	2-18
2-10	Output File Format	2-20
3-1	Structure of Program MODIFY	3-2
3-2	Data Modification Subroutines Used By Program MODIFY	3-4
3-3	Flowchart for Program MODIFY	3-6
3-4	Flowchart of Subroutine COMSYS	3-11
3-5	Flowchart of Subroutine LDFACT	3-14
3-6	Flowchart of Subroutine NOISE	3-15
5-1	Organization of COMPUTE	5-2
5-2	Flowchart for the Input and Initialization Module	5-3
5-3	Flowchart for the Traffic Flow Module	5-4
5-4	Flowchart for the Operating Statistics Module	5-5

## EXHIBITS

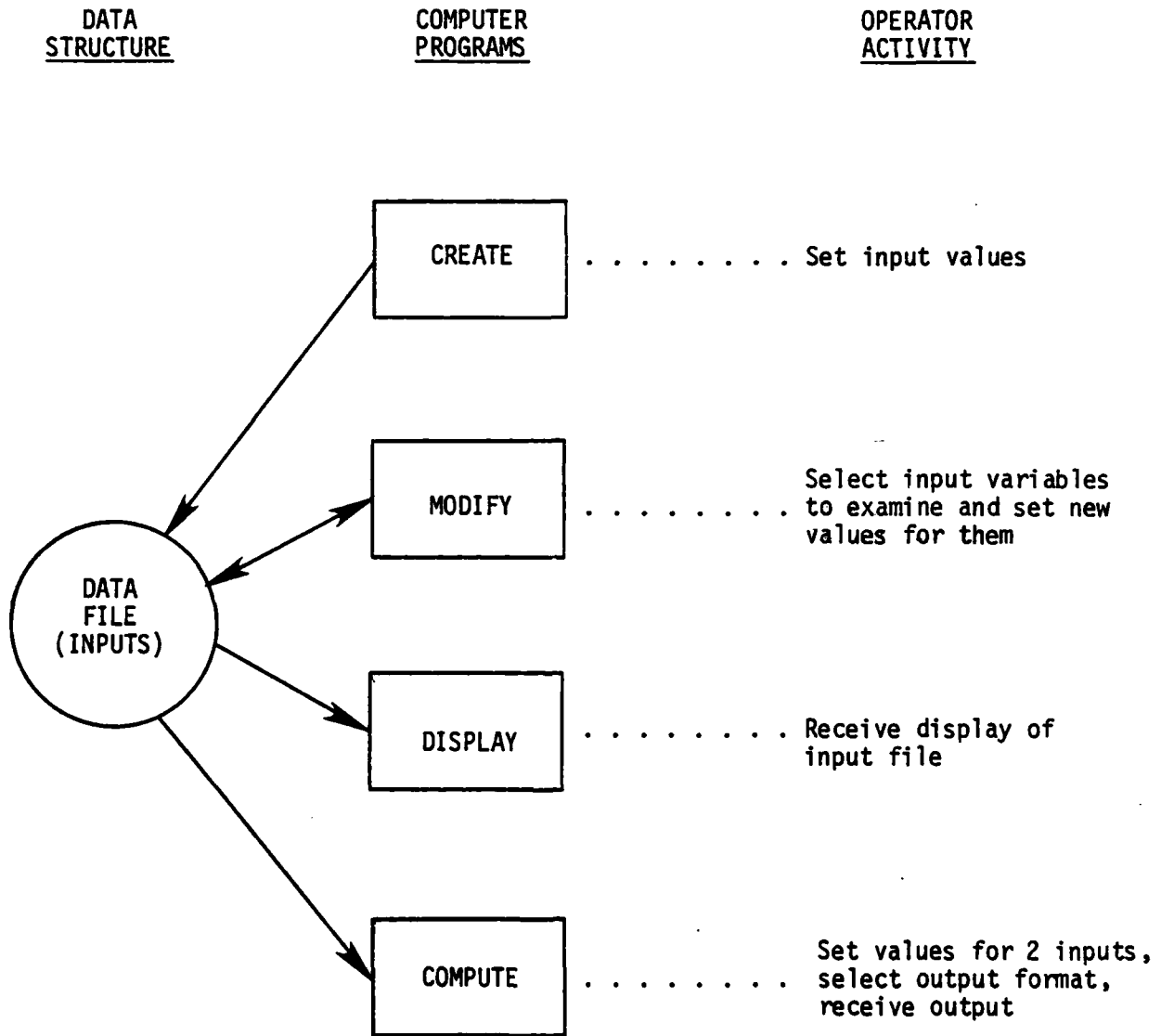
<u>Number</u>	<u>Title</u>	<u>Page</u>
5-5	Flowchart for Segment A of the Operating Statistics Module	5-6
5-6	Flowchart for Segment B of Operating Statistics Module	5-7
5-7	Flowchart for Subroutine TRANS	5-8
5-8	Flowchart for the Performance Measure Module	5-10
5-9	Flowchart for the Output Module	5-11
5-10	Variables in COMPUTE	5-12
5-11	Logical I/O Unit Assignments	5-18

## 1.0 INTRODUCTION

↓  
The purpose of this volume is to document the computer programs and data structures in the TOS analysis package in sufficient detail to enable a programmer to modify the package or transfer it between computer systems. The volume is organized into five chapters and three appendices. The remainder of this chapter discusses the overall organization of the analysis package. Chapters 2.0 through 5.0 discuss, in turn, each of the four computer programs in the package and present flowcharts, descriptions of the variables and subroutines, and discussions of the code. Appendix A contains a listing of the four programs. Appendix B contains a variable name cross-reference dictionary. Appendix C contains a statement label dictionary. All programs are written in standard FORTRAN IV and contain internal commentary. They are currently implemented on the Amdahl 470/V7 computer at The University of Michigan.

The analysis package performs three basic functions: (1) interacting with the operator to set, alter, or examine the values of the inputs; (2) performing the computations specified by the mathematics of the model; and (3) displaying the outputs. The package consists of four computer programs and one external data structure. The organization of the package, the interactions between the programs and the data structure, and the operator activities are shown in exhibit 1-1. Program CREATE prompts the operator to specify values for all of the inputs in the data file and uses them to create a new data file. Program MODIFY reads an existing data file, and allows the user to select data elements

## EXHIBIT 1-1: ANALYSIS PACKAGE ORGANIZATION



for display or modification. The program then creates a new data file. Program DISPLAY reads, formats, and displays an existing data file. Program COMPUTE performs all of the computations required by the model, allows the operator to select the output format, and displays the outputs.

## 2.0 PROGRAM CREATE

Program CREATE was written to facilitate construction of the data file which contains input data for the TOS simulation. Program CREATE is written in FORTRAN IV and is set up to be run interactively from a terminal. The programming code and logic are straightforward and should facilitate implementation on any computer the potential user desires. The remainder of this section discusses the variables and characteristics of program CREATE.

### 2.1 VARIABLES IN PROGRAM CREATE

Exhibits 2-1 through 2-7 contain all of the variables used in program CREATE, indicate the type of variable as implemented in the program, and give a definition of the meaning of each variables. The contents of these exhibits are:

- Exhibit 2-1: Network Configuration Data
- Exhibit 2-2: Communications System Data
- Exhibit 2-3: Message Types Data
- Exhibit 2-4: Route Cross Message Data
- Exhibit 2-5: Engineering Data
- Exhibit 2-6: Error Rate Data
- Exhibit 2-7: Program Execution Controls

The variables which contain information about the TOS system have names mnemonically derived so that understanding of the meaning of the variables is facilitated. The exceptions to this general rule are the logic variables which are used to control the program flow.

## EXHIBIT 2-1: CONFIGURATION VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
NBDE	I*4	The number of brigades
BDNAME(I)	R*8	The name of Brigade I
NBN(I)	I*4	Number of battalions in Brigade I
BD(I)	I*4	Type of processor at Brigade I: = 1 if TCS, = 2 if TCT
NLINC(I)	R*8	The name of the channel connecting Brigade I with the FEP
BNNAME(I,J)	R*8	The name of Battalion J of Brigade I
BN(I,J)	I*4	The type of processor at Battalion J of Brigade I: = 1 if TCS, = 2 if TCT
NLINK(I,J)	R*8	The name of the channel connecting Battalion J to Brigade I
NOU	I*4	The number of Other TOS Users (All users which are not either brigades or their subordinate battalions are considered Other TOS users.)
OUNAME(I)	R*8	The name of Other TOS User I
OU(I)	I*4	Type of processor at Other TOS User I: = 1 if TCS, = 2 if TCT
NLINCK(I)	R*8	Name of the channel connecting Other TOS User I to the FEP
NC	I*4	The number of communication channels in the TOS network
CNAM(I)	R*8	The name of TOS communication channel I

-- Continued --

EXHIBIT 2-1: CONFIGURATION VARIABLES IN CREATE  
(Continued)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
NR	I*4	The number of routes in the TOS communication network: equals the number of Other TOS users plus number of brigades plus number of battalions
RP(I,1)	I*4	The number of the route. Routes are numbered in the same order as given for the route name. (The following order of routes is followed throughout the TOS analysis package: routes numbered 1 through NOU are the Other TOS Users; routes NOU+1 to NOU+NBDE are Brigades 1 through NBDE, next come the battalions, first the battalions of Brigade 1 then the battalions of 2, etc.)
RNAM(I)	R*8	The name of route I.
RP(I,2)	I*4	The node with which node I communicates. By convention this equals 0 (the FEP) for Other TOS Users and Brigades. For a Battalion, it equals the route number of its parent brigade
NL	I*4	Number of links, equals the number of routes
LP(I,1)	I*4	Equal to RP(I,1), the number of the link
LP(I,2)	I*4	Number of channel connecting node I with the FEP, if I is a brigade or Other TOS User, or with the parent brigade if I is a battalion
LP(I,3)	I*4	The number of the node with which node I communicates. If node I communicates with the FEP, LP(I,3) equals NR plus 1, if node I is a battalion, LP(I,3) is the number of its parent brigade

-- Continued --

EXHIBIT 2-1: CONFIGURATION VARIABLES IN CREATE  
(Concluded)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
LINC(I)	I*4	The number of the channel connecting Brigade I with the FEP
LINK(I,J)	I*4	The number of the channel connecting Battalion J with Brigade I
		LINCK(I) I*4 The number of the channel connecting Other TOS User I with the FEP
IPD(I)	I*4	The type of processor at node I: = 1 if TCS, = 2 if TCT
NP	I*4	Number of processors: equal NR+1
NN	I*4	Number of nodes: equals NP + NC + 3

EXHIBIT 2-2: COMMUNICATIONS SYSTEM VARIABLES  
IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
CD(I,1)	R*4	Rise time, in milliseconds, of Channel I
CD(I,2)	R*4	Transmission rate, in characters per second, of Channel I
CD(I,3)	R*4	Proportion of time Channel I is being used for voice communications
CD(I,4)	R*4	Number of lines on Channel I

## EXHIBIT 2-3: MESSAGE VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
NM	I*4	Total number of message types in the system
NUM	I*4	Number of user-initiated messages in the TOS system (By implication, the number of TOS-generated messages is NM - NUM)
MNAM(I)	R*8	The name of Message Type I. (By convention user-initiated messages are numbered 1 through NUM and TOS-initiated messages are numbered NUM + 1 through NM)
MD(I,1)	R*4	The average length in characters of Message Type I
IMP(I,1)	R*8	The name of message type caused as output on the same route by Message Type I: = 0, if none
IMP(I,2)	R*8	The name of message type caused as output on another route by Message Type I: = 0, if none or if Message I is a TOS-generated message
MP(I,1)	I*4	The number of Message Type IMP(I,1)
MP(I,2)	I*4	The number of Message Type IMP(I,2)

EXHIBIT 2-4: ROUTE CROSS MESSAGE ARRAY VARIABLES IN CREATE  
(I = Route Number, J = Message Number)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
(If $J \leq \text{NUM}$ , Message J is a user-initiated message)		
RM(I,J,1)	R*4	Initiation rate, in messages per hour, of Message J at Node I
RM(I,J,2)	R*4	= 0, not used for user-initiated messages
RM(I,J,3)	R*4	= 0 if I is not a battalion. Equals the probability that Message J is deleted during hierarchical review at the parent brigade if I is a battalion
RM(I,J,4)	R*4	= 0 if I is not a battalion. Equals the probability that Message J is altered, but not deleted, during hierarchical review at the parent brigade if I is a battalion
(If $J > \text{NUM}$ , Message J is a TOS-initiated message)		
RM(I,J,1)	R*4	= $\alpha(I,J)$ , proportionality constant used to determine the initiation rate of TOS-initiated Message J on route I
RM(I,J,2)	R*4	= $\beta(I,J)$ , proportionality constant used to which determine the initiation rate of TOS-initiated Message J on route I
RM(I,J,3)	R*4	= 0, not used for TOS-initiated messages
RM(I,J,4)	R*4	= 0, not used for TOS-initiated messages

## EXHIBIT 2-5: ENGINEERING DATA IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
DD(1)	R*4	Message Disk Mean access time in milliseconds
DD(2)	R*4	Data Disk Mean access time in milliseconds
GPD(1,1)	R*4	CPU, in milliseconds per message, required to originate a message, at a TCS
GPD(1,2)	R*4	CPU, in milliseconds per message, required to originate a message at a TCS
GPD(2,1)	R*4	CPU, time in milliseconds per message, required to send the message, at a TCS
GPD(2,2)	R*4	CPU, time in milliseconds per message, required to send the message at a TCS
GPD(3,1)	R*4	CPU, time in milliseconds per message, required to receive a message at a TCS
GPD(3,2)	R*4	CPU, time in milliseconds per message, required to receive a message, at a TCS
GPD(4,1)	R*4	CPU, time in milliseconds per message, to template check a message, at a TCS
GPD(4,2)	R*4	CPU, time in milliseconds per message, required to template a check a message, at a TCS
GPD(5,1)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(5,2)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS

-- Continued --

EXHIBIT 2-5: ENGINEERING DATA IN CREATE  
(Continued)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
GPD(6,1)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(6,2)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(7,1)	R*4	CPU, time in milliseconds per message, required to originate a message, at a TCS
GPD(7,2)	R*4	CPU, time in milliseconds per message, required to originate a message, at a TCS
GPD(8,1)	R*4	CPU, time in milliseconds per message, required to send a message, at a TCS
GPD(8,2)	R*4	CPU, time in milliseconds per message, required to send a message, at a TCS
GPD(9,1)	R*4	CPU, time in milliseconds per message, required to send to receive a message, per message
GPD(9,2)	R*4	CPU, time in milliseconds per message, required to receive a message, at a TCS
GPD(10,1)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(10,2)	R*4	CPU, time in milliseconds per message, required to terminate a message, at a TCS
GPD(11,1)	R*4	CPU, time in milliseconds per message, required to initiate a message, at a TCS
GPD(11,2)	R*4	CPU, time in milliseconds per message, required to initiate a message at a TCS

-- Continued --

EXHIBIT 2-5: ENGINEERING DATA IN CREATE  
(Concluded)

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
MD(I,2)	R*4	Data base processing time, in milliseconds, of Message Type I
MD(I,3)	R*4	Number of message disk reads or writes for Message Type I
MD(I,4)	R*4	Number of data disks reads or writes for Message Type I
PD(I)	I*4	Number of templates at node I
MD(I,4)	R*4	Number of data disks reads or writes for Message Type I
PD(I)	I*4	Number if templates at node I: = 0 if node I is not a Brigade
PD(NR+1)	I*4	Number of templates at the FEP

## EXHIBIT 2-6: ENVIRONMENTAL VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
LD(I,1)	R*4	The bit error rate for a message sent from Node I to the next node, in bits in error per 1000 bits
LD(I,2)	R*4	The bit error rate for a message sent to Node I, in bits in error per 1000 bits

## EXHIBIT 2-7: LOGIC VARIABLES IN CREATE

<u>Name</u>	<u>Type</u>	<u>Meaning</u>
IV	I*4	= 1 if user desires short prompts
ITCT	I*4	= 1 if a TCT is present in the TOS system
ITCT	I*4	= 2 if a TCT is present in the TOS system
IST	I*4	A temporary variable to determine if TCSs and TCTs are present
TCS	I*4	Literally, the character string "TCS"
TCT	I*4	Literally, the character string "TCT"
IFG	I*4	= 1 if the Route cross Message array is to be read in later
IW	I*4	= 1 if the user desires to call the display routine to print the data on the terminal
IWF	I*4	= 1 if the user desires to create an output file

(I, II, III, IJ, I1, I2, J, JI, JJ, JJJ, JK, JN, JR, K, RNUM are all indices used in the CREATE logic)

A convention for numbering the users of the TOS system is followed throughout all four of the computer programs. In this system, the first users are always the non-brigade users of the TOS system. Immediately following these are all of the brigade users. Following the brigades are their battalion users: all the battalions of the first brigade, then all the battalions of the second brigade, etc. In this way, the type of TOS user can be inferred by its number without having to append additional identifiers to the number or names. The use of this convention will become obvious when the data file is discussed.

The arrays BDNAM, BNNAME, OUNAM, CNAM, RNAM, and MNAM are declared as R\*8 variables in program CREATE. They are, however, used as names in CREATE and may contain strings of up to eight alpha-numeric characters. They are included as a convenience to the user and are not used in the logic of any of the programs in the TOS analysis package.

It should be noted that some of the variables in CREATE are redundant. For example, the LP and RP arrays contain some of the same information. This organization facilitated the coding of program COMPUTE.

## 2.2 BASIC FLOW OF PROGRAM CREATE

Exhibit 2-8 contains a basic flowchart of program CREATE. The steps in CREATE are implemented in a straightforward serial fashion. They are set up so that succeeding steps use information which are input in earlier steps. This information is used to control the internal DO Loops so they provide the correct number of prompts.

## EXHIBIT 2-8: BASIC FLOW OF CREATE

- Step 0: Set IV for long or short prompts IV = 1 short
- Step 1: Brigade Information  
 Enter number of Brigades: NBDE  
 For each Brigade, enter name, type of processor, and  
 number of Battalions: BDNAM(I), BD(I), NBN(I)  
 Call CHECK for BD(I) to check name
- Step 2: Battalion Information  
 For each Battalion, enter name and type of processor  
 BNNAME(I,J), BN(I,J) ((I,J) = Battalion J of Brigade I)  
 Call CHECK for BN(I,J) to check name
- Step 3: Other TOS User information  
 Enter number of Other TOS Users: NOU  
 For each Other TOS User, enter name and type of processor  
 OUNAM(I), OU(I)  
 Call CHECK for OU(I) to check name
- Step 4: Channel Data  
 Enter number of communications channels: NC  
 For each channel, enter the name CNAM(I)
- Step 5: Connect User with Channels  
 For each Battalion, enter name of channel connecting it to its  
 parent Brigade, NLINK(I,J)  
 Call CHECK 1 to check name and assign channel number to  
 LINK(I,J)  
 For each Brigade enter name of channel connecting it to the  
 FEP: NLINC(I)  
 Call CHECK1 to check name and assign channel number to  
 LINC(I)  
 For each Other TOS User, enter name of channel connecting it to  
 the FEP: NLINCK(I)  
 Call CHECK1 to check name and assign channel number to  
 LINCK(I)
- Step 6: Make Configuration Assignments  
 Compute NL, NR, NP, NN  
 For each TOS user, assign values to RNAM(I), RP(I,1), RP(I,2),  
 LP(I,1), LP(I,2), LP(I,3), IPD(I)  
 Check to determine if configuration includes both TCSs and TCTs

-- Continued

# EXHIBIT 2-8: BASIC FLOW OF CREATE (Continued)

- Step 7:**      **Message Data**  
Enter number of messages and number of user initiated messages,  
NM, NUM  
For each user-initiated message enter name: MNAM(I)  
For each TOS-initiated message enter name: MNAM(I)  
For each user-initiated message enter mean length,  
name of message caused as output on same route, and name of  
message caused as output on other routes: MD(I,1), IMP(I,1),  
IMP(I,2)  
Call CHECK2 to check message names and assign message numbers  
to MP(I,1) and MP(I,2)  
For each TOS-initiated message, enter mean length: MD(I,1)  
Assign MP(I,1) = MP(I,2) = 0
- Step 8:**      **Route Cross Message Array**  
(If Array to be inserted into file later, set IFG=1 and bypass  
step 8)  
For each Other TOS User, then for each Brigade, enter for each  
user-initiated message the message initiation rate:  
RM(I,J,1). Then assign RM(I,J,2) = RM(I,J,3) = RM(I,J,4)  
= 0  
For each TOS-initiated message enter proportionality  
constants  $\alpha(I,J)$  and  $\beta(I,J)$ , RM(I,J,1) and RM(I,J,2).  
Then assign RM(I,J,3) = RM(I,J,4) = 0  
For each Battalion, for each user-initiated message, enter the  
message initiation rate, the probability that the message is  
deleted at Brigade review and the probability the message is  
altered at review: RM(I,J,1), RM(I,J,3), and RM(I,J,4).  
Call CHECK 3 to insure probabilities are in the interval  
0-1; then assign RM(I,J,2) = 0  
For each TOS initiated message, enter proportionality  
constants  $\alpha(I,J)$  and  $\beta(I,J)$ , and RM(I,J,2); then assign  
RM(I,J,3) = RM(I,J,4) = 0
- Step 9:**      **Engineering Data**  
Enter data disk and message disk access times: DD(1), and  
DD(2)  
Enter TCS data: GPD(1...5, 1 and 2)  
Enter TCT data: GPD(7...10, 1 and 2)  
Enter FEP data: GPD(11, 1 and 2), and GPD(6, 1 and 2)  
For each channel, enter rise time, transmission rate, and  
number of lines: CD(I,1), CD(I,2), and CD(I,4)  
For each message, enter data base processing time, number of  
message disk reads or writes, and number of data disk reads  
or writes: MD(I,2), MD(I,3) and MD(I,4)  
For each Brigade enter number of templates: PD(I)  
Enter number of templates at FEP: PD(NR+1)  
Assign PD(I) = 0 for all nodes not a Brigade or the FEP
- Step 10:**      **Voice Competition Data**  
For each channel, enter proportion of time channel is used for  
voice transmission: CD(I,3)  
Call CHECK4 to insure CD(I,3) is in the range 0-1

-- Continued --

# EXHIBIT 2-8: BASIC FLOW OF CREATE (Concluded)

- STEP 11:      Error Rate Data  
               For each Other TOS User, then for each Brigade:  
                   enter the bit error rate for a message from the user to the  
                   the FEP and the bit error rate for a message from the FEP to  
                   the user: LD(I,1), and LD(I,2)  
               Call CHECK 3 to ensure LD(I,1) and LD(I,2) are in the  
                   interval 0-1  
               For each Battalion, enter the bit error rate for a message sent  
                   to the parent Brigade and the bit error rate for a message to  
                   the Battalion.  
               Call CHECK 3 to ensure LD(I,1) and LD(I,2) are in the  
                   interval 0-1.
- Step 12:      Display Data  
               If user wishes to display data, enter IW=1  
               If IW=1, call DISP, if IW  $\neq$  1 bypass step 12
- STEP 13:      Write Output File  
               If user wishes to create an output file, enter IFW=1  
               If IFW  $\neq$  1, bypass step 13.  
               If IFW=1, Write counting indices    NR, NM, NN, NUM, NC, NP, NL,  
                   NOU, NBDE, and IFG  
               For each route write RNAM(I), RP(I,1), RP(I,2), and IPD(I)  
               For each message write MNAM(I), MP(I,1) MP(I,2), MD(I,1)  
                   MD(I,2), MD(I,3), and MD(I,4)  
               For each channel, write CNAM(I), CD(I,1), CD(I,2), CD(I,3), and  
                   CD(I,4)  
               For each processor, write PD(I)  
               For each link, write LP(I,1), LP(I,2), LP(I,3), LD(I,1), and  
                   LD(I,2)  
               For each route, for each message, write RM(I,J,1), RM(I,J,2),  
                   RM(I,J,3), and RM(I,J,4)  
                   (or write a line of "\*" if array to be input later)  
               Write DD(1), and DD(2)  
               Write GPD(I,1) and GPD(I,2), for I=1,...,11
- Step 14:      Terminate

Also included in the internal logic of each of the steps is a means of prompting using either short or long forms of the prompts. If the short form is elected in step 0, then in each succeeding step the first of a series of similar prompts will contain a long, sentence-like prompt. All subsequent prompts for similar data will be given in short form only. This was instituted to decrease the running time of program CREATE.

### 2.3 SUBROUTINES IN CREATE

Exhibit 2-9 lists the six subroutines which are part of program CREATE and indicates their basic functions. Five of the six subroutines are used for error checking during the course of using the program CREATE. The sixth subroutine permits display of the input data in an easy to read format. Each of these subroutines is discussed in greater detail in the following paragraphs.

Subroutine CHECK compares the name input by the user for the processor at a node against the symbols "TCT" and "TCS". If the processor type which is input by the user does not correspond to one of these two names, CHECK will so indicate to the user and prompt for the correct name. The checks and prompts will continue until the user inputs either "TCT" or "TCT".

Subroutine CHECK1 compares the channel name which the user has input as the channel connecting two nodes with the list of channel names which was input earlier. If the name the user has input is not contained in the previously defined list, CHECK1 will so indicate to the user and request another name to be input. This prompting will continue until a proper channel name is input. When CHECK1 sees a proper channel name, it will assign and return the number of that channel to the main program where channel numbers, not names, are used for identification.

## EXHIBIT 2-9: SUBROUTINES IN CREATE

<u>Name</u>	<u>Function</u>
CHECK	Checks processor type
CHECK1	Checks channel name, assigns channel number
CHECK2	Checks message name, assigns message number
CHECK3	Checks probability range
CHECK4	Checks proportion range
DISP	Displays data in easy to read format

Subroutine CHECK2 performs much the same function as subroutine CHECK1. Here the name of the message caused as an output by a particular message is compared with the previously input list of message names. If the name is not contained in that list, the user is prompted to enter another name until a legal message name is seen. When this occurs, subroutine CHECK2 will determine the number of the message input and return that number to the main program.

During the course of CREATE, the user is prompted for the probabilities of various actions. When the user has answered such a prompt, the number is checked in the main program to see if it is in the range 0-1. If the input number is outside this range, CHECK3 is called and the user is prompted to enter a number in the range 0-1.

Subroutine CHECK4 is identical to subroutine CHECK3 except it is used to check the input values for proportions rather than probabilities. If the input proportion is outside the range of 0-1, CHECK4 is called.

When all the data have been input, the user is presented with the instruction, "If you want to display the output file, enter a 1". If the user enters a 1, subroutine DISP is called and the data are printed at the terminal in an easy to read format. Subroutine DISP is essentially program DISPLAY; however, the data are passed as subroutine arguments rather than being read in from an input file, as is the case with program DISPLAY.

## 2.4 DATA FILE

Exhibit 2-10 contains a description of the data file which is the output of program CREATE. The variable names are as defined in exhibits 2-1 through 2-7 in which the formats of the lines in the output files are

## EXHIBIT 2-10: OUTPUT FILE FORMAT

<u>Line</u>	<u>Elements</u>	<u>Format</u>
COUNTERS		
1	NR, NM, NN, NUM, NC, NP, NL, NOU, NBDE, IFG	10(I3, 1X)
ROUTE INFORMATION		
2 thru NR+1	RNAM(I), RP(I,1), RP(I,2), IPD(I)	A8, 3(1X,I3)
MESSAGE INFORMATION		
NR+2 thru NR+NM+1	MNAM(I), MP(I,1), MP(I,2), MD(I,1), MD(I,2), MD(I,3), MD(I,4)	A8, 2(1X,I3), 4(1X, E9.3)
CHANNEL INFORMATION		
NR+NM+2 thru NR+NM+NC+1	CNAM(I), CD(I,1), CD(I,2), CD(I,3), CD(I,4)	A8, 4(1X,E9.3)
NUMBER OF TEMPLATES		
NR+NM+NC+2 thru NR+NM+NC+NP+1	PD(I)	2X, I3
LINK DATA		
NR+NM+NC+NP+2 thru NR+NM+NC+NP+NL+1	LP(I,1), LP(I,2), LP(I,3), LD(I,1), LD(I,2)	3(I3, 1X), 2(E12.6, 1X)

# EXHIBIT 2-10: OUTPUT FILE FORMAT (Concluded)

## ROUTE CROSS MESSAGE ARRAY (If IFG=1 the array is a line of 50 "s")

NR+NM+NC+NP+NL+2	}			
thru				
NR+NM+NC+NP+NL+NM+1	}			
thru				
NR+NM+NC+NP+NL+(NR-1)*NM+2	}			
thru				
NR+NM+NC+NP+NL+NR*NM+1	}			

RM(1,J,1), RM(1,J,2), RM(1,J,3), RM(1,J,4) 4(E9.3,1X)

RM(NR,J,1), RM(NR,J,2), RM(NR,J,3), RM(NR,J,4) 4(E9.3,1X)

## DISK ACCESS TIMES

NR+NM+NC+NP+NL+NR*NM+2	DD(1), DD(2)	2(E9.3, 1X)
------------------------	--------------	-------------

## GENERAL PROCESSOR DATA

NR+NM+NC+NP+NL+NR*NM+3	}			
thru				
NR+NM+NC+NP+NL+NR*NM+13	}			

GPD(1,1), GPD(1,2) 2(E9.3,1X)

also shown. This output file is set up so that the first line contains the counters which are used to determine the number of lines in the different sections of the data file. For example, NR indicates the number of lines in the Route Information section; NM, the number of lines in the Message Information section; etc. The file is implemented in this manner so that the controls for reading the file would be contained within the file itself and no other information need be given. This data file is in a format compatible with the other three programs in the TOS analysis package.

## 2.5 CODE OF PROGRAM CREATE

Program CREATE is written in FORTRAN IV using library subroutines existing on the Amdahl 460/V7 computer at The University of Michigan. The code is liberally interspersed with comment statements which indicate the function of the various parts of the program.

The initial version of program CREATE was a short, simple program which would suffice for most applications; however, during the checkout of the other programs in the analysis package, it was obvious that additional logic needed to be included in CREATE to permit its use with various test cases. In its simplest form, CREATE was implemented assuming a standard division configuration which had more than one brigade, more than one battalion for each brigade, and more than one other TOS user. In this case, simple DO LOOPS will suffice; however, test cases were implemented which contained only one or two nodes and in those situations additional logic was required to bypass unnecessary DO LOOPS. Thus, there are a large number of "IF" comparisons in CREATE. These are checks to bypass portions of code which are not needed.

The evolution of program CREATE included a decision to implement a series of short prompts as part of the program. There are essentially two sets of queries contained in CREATE. One set contains long, sentence-like queries. In the other set the queries are short and are used to prompt for subsequent entries of a similar type. Thus, there are also numerous logic checks to determine when these short prompts are to be called. This also necessitated additional logic to account for the various test cases which were to be implemented, although most of that logic would not be exercised if the program is used for the input of a standard division configuration.

All of the input statements in program CREATE are free format and make use of the library program, FREAD, which is available on the Amdahl computer on which program CREATE was written. These free format read statements were implemented to make the user's task of inputting the data easier. If program CREATE is implemented on a different computer system, the user must determine if FREAD is available. If FREAD is not available, the input statements need to be replaced with either conventional formatted statements or with whatever free format subroutine is available on the host machine.

### 3.0 PROGRAM MODIFY

Program MODIFY was written to facilitate alteration of an existing data file. The program is written in FORTRAN IV and is designed to be run interactively from a terminal. This section provides information which may be of use to a programmer who wishes to modify the logic of this program or to transfer this program to a new computer system. Section 3.1 contains a description of the structure of the program MODIFY and provides a list of the subroutines which are used. Section 3.2 contains a discussion of the code and how to modify the existing code within the framework of the structure described in section 3.1. Section 3.3 contains flowcharts of the main program and subroutine.

#### 3.1 Structure of Program MODIFY

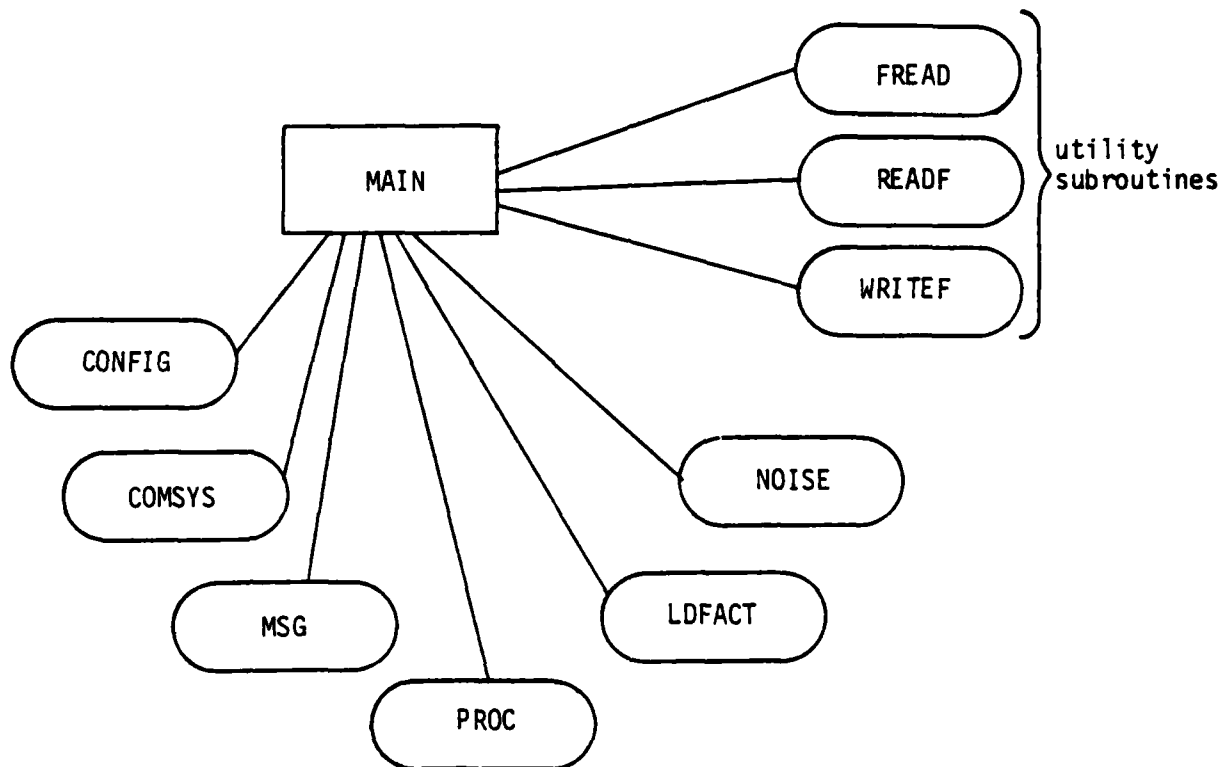
Program MODIFY is implemented as a main executive program which calls several subroutines to perform the actual modification of input data. Exhibit 3-1 illustrates the general structure of the program.

The data file for the TOS analysis package consists of data of six logical types. Exhibits 2-1 through 2-6 list the data by type. This classification of data by type is reflected in the structure of program MODIFY. The main executive program calls six subroutines to perform modification of data, one for each type of data. Exhibit 3-2 relates these six subroutines to the type of data which they modify.

Program MODIFY also uses three utility subroutines to perform I/O functions. Subroutine READF reads an existing data file from I/O

## EXHIBIT 3-1: STRUCTURE OF PROGRAM MODIFY

Main Executive Program and Subroutines



unit 1. Subroutine WRITEF writes a new data file onto I/O unit 2. Subroutine FREAD is a general free-format read routine which is discussed in section 2.3.

### 3.2 CODE OF PROGRAM MODIFY

The code of program MODIFY was produced using a top-down approach. The result is a modular program which consists of a short executive main program and a number of specialized subroutines. Section 3.1 describes the structure of program MODIFY and lists its subroutines. A detailed flowchart of program MODIFY is found in section 3.3.

Three utility subroutines are used to perform I/O functions. READF reads the data file from I/O unit 1. This routine was created by copying the code used to perform the same function in program DISPLAY. WRITEF writes the new data file to I/O unit 2 and was created by copying code used by program CREATE. The free-format read subroutine, FREAD, is a library program on the Amdahl computer at The University of Michigan.

The actual modification of data is performed by the six subroutines listed in exhibit 3-2. The six subroutines correspond to the six types of data in the data file. The coding of these subroutines was performed to allow modification of those data which were changed in the course of the analysis of the TOS communications network; however, the program does not currently allow modification of all of the data items.

Subroutine COMSYS allows the user to modify and examine any of the communication system data. Subroutine LDFACT allows modification of some of the route cross message data. The user is allowed to multiply message traffic rates for the entire communications network by a factor. Subroutine NOISE allows the user to set bit error rates by channel, and to

## EXHIBIT 3-2: DATA MODIFICATION SUBROUTINES USED BY PROGRAM MODIFY

<u>Subroutine</u>	<u>Type of Data Modified</u>
CONFIG	System Configuration
COMSYS	Communications System
MSG	Message Data
PROC	Processor Data
LDFACT	Route Cross Message Data
NOISE	Bit Error Rates

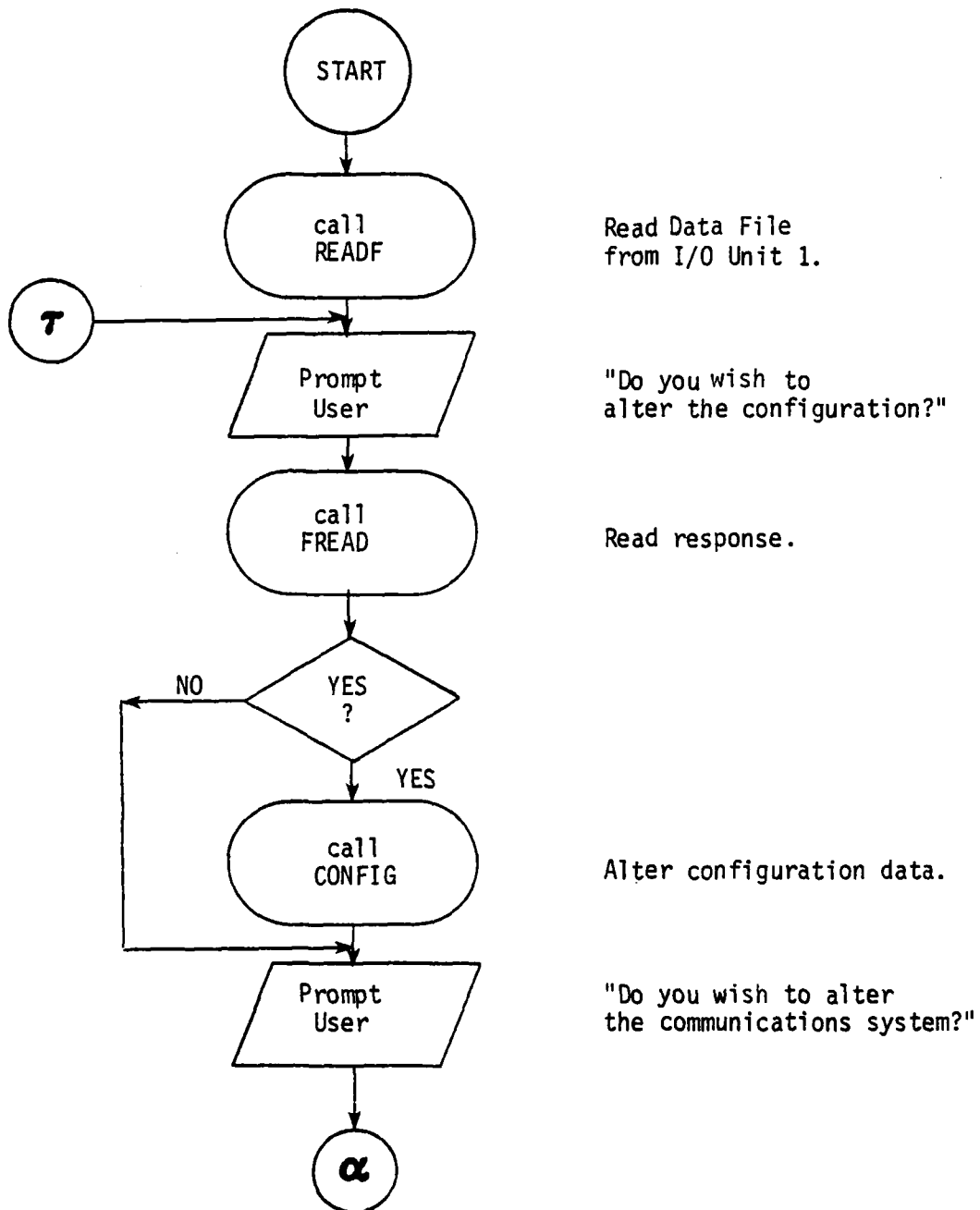
display bit error rates. All of the code which displays input data was copied from program DISPLAY. Subroutines CONFIG, MSG, and PROC are dummy routines, since it was not necessary, in the course of the analysis of TOS, to use program MODIFY to change the values of any of the configuration, message, or processor data.

In order to extend the range of the current version of MODIFY to allow modification of other data items, the programmer may write new versions of some or all of the six special subroutines which perform data modification. Some changes would be straightforward. For example, it would be easy to write into subroutine PROC the ability to replace most of the processor data; however, some other modifications are more complex, and could involve the creation of interdependence among the six subroutines. For example, the insertion or deletion of nodes in CONFIG, which handles configuration data, may necessitate insertion, deletion, or adjustment of communication system data, message data, route cross message data, and error rate data. Such changes should preferably be handled by requiring the user to run program CREATE to create a new data file. The current version of MODIFY is designed to replace existing values in the input data file, not to insert or delete data.

### 3.3 FLOW CHARTS OF THE PROGRAM MODIFY AND SELECTED SUBROUTINES

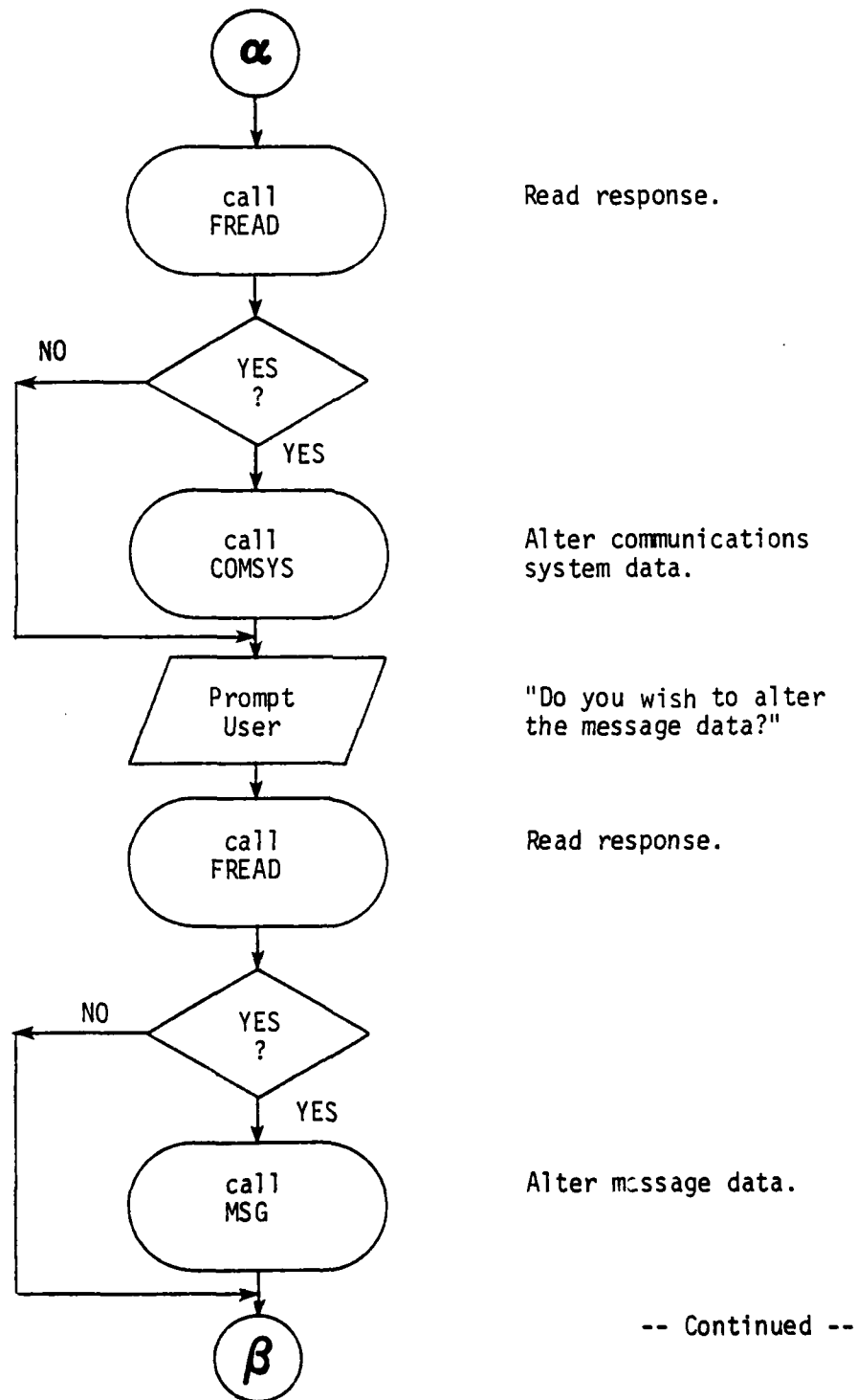
Exhibits 3-3 through 3-6 contains detailed flowcharts of the main program and subroutines COMSYS, LDFACT, and NOISE.

EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY

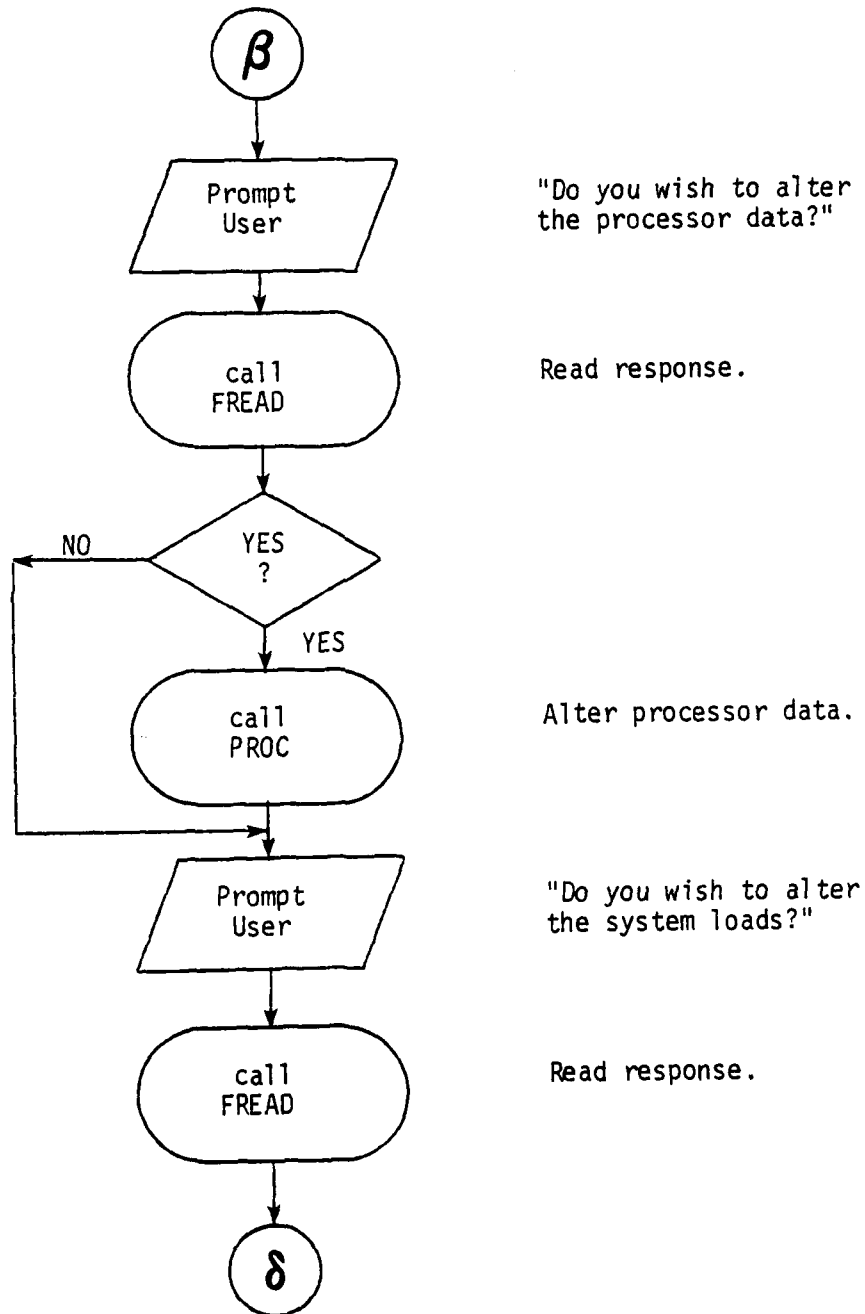


-- Continued --

EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Continued)



## EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Continued)



-- Continued --

EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Continued)

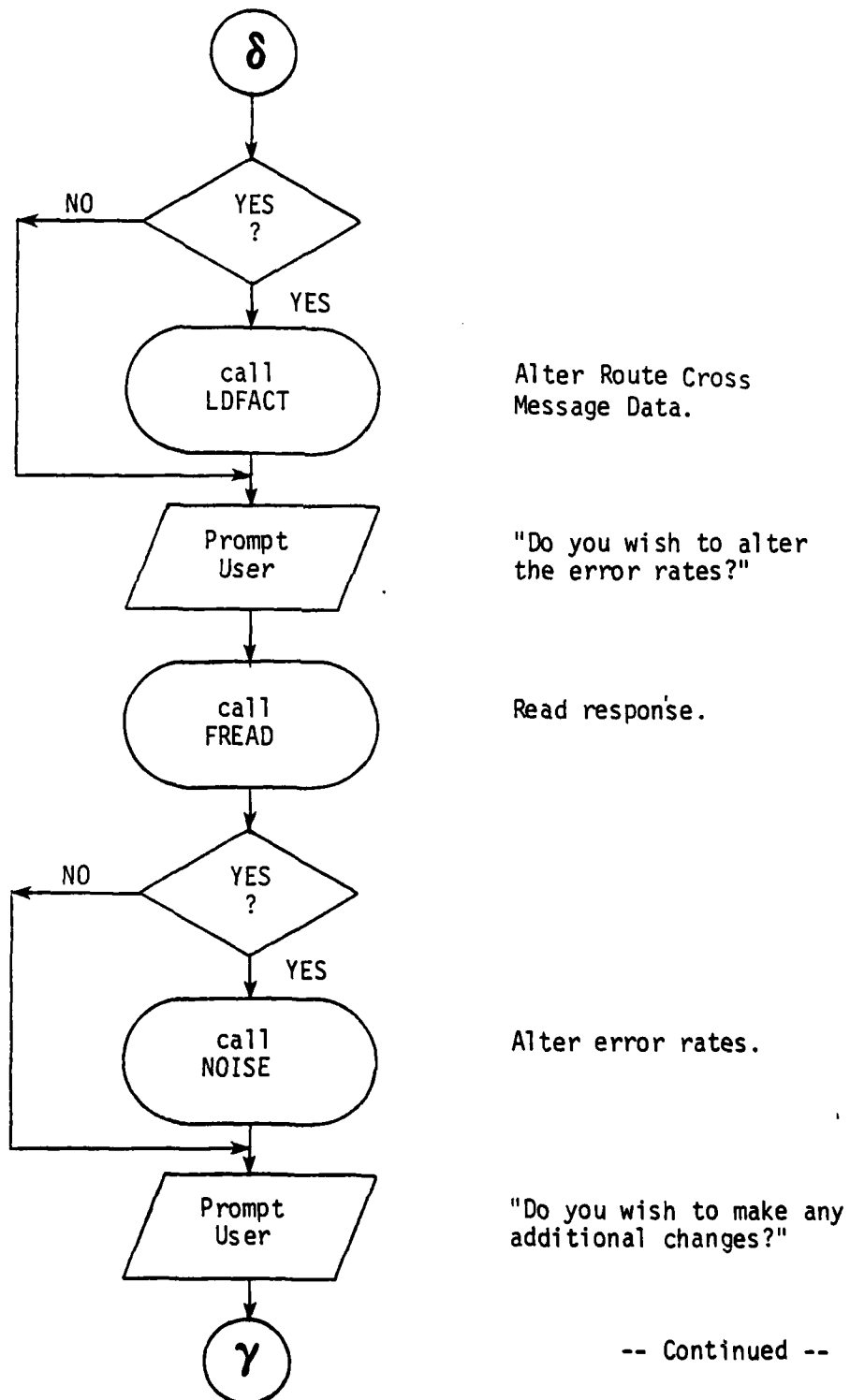


EXHIBIT 3-3: FLOWCHART FOR PROGRAM MODIFY (Concluded)

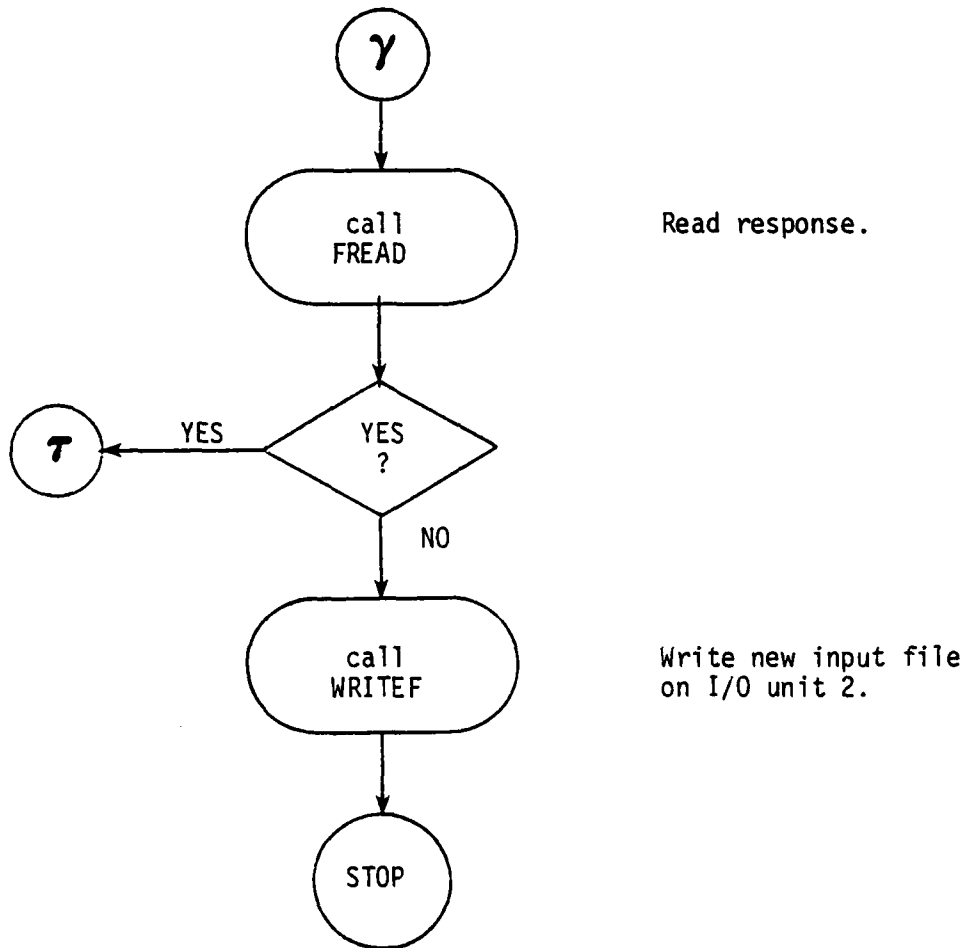


EXHIBIT 3-4: FLOWCHART OF SUBROUTINE COMSYS

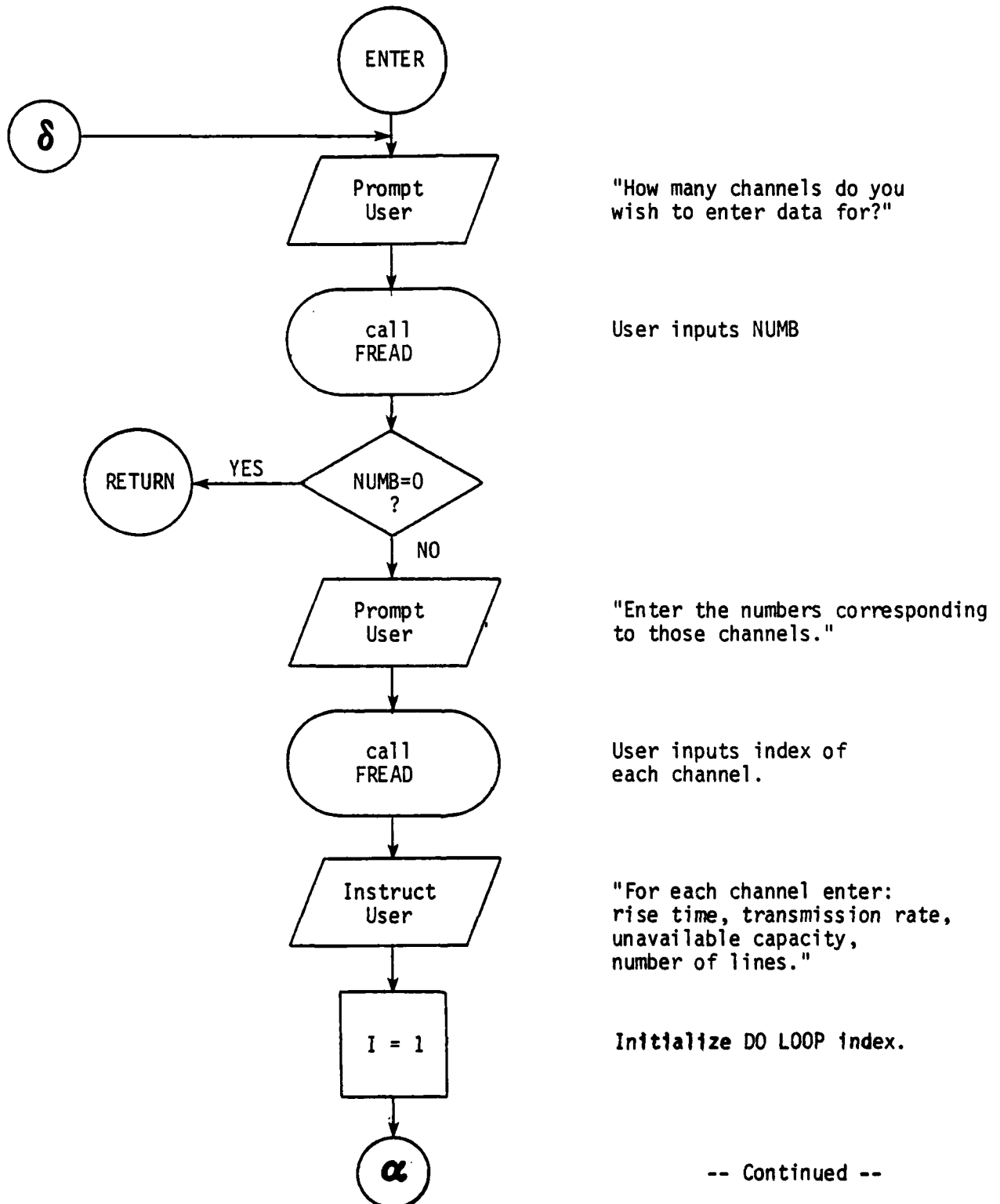


EXHIBIT 3-4: FLOWCHART OF SUBROUTINE COMSYS (Continued)

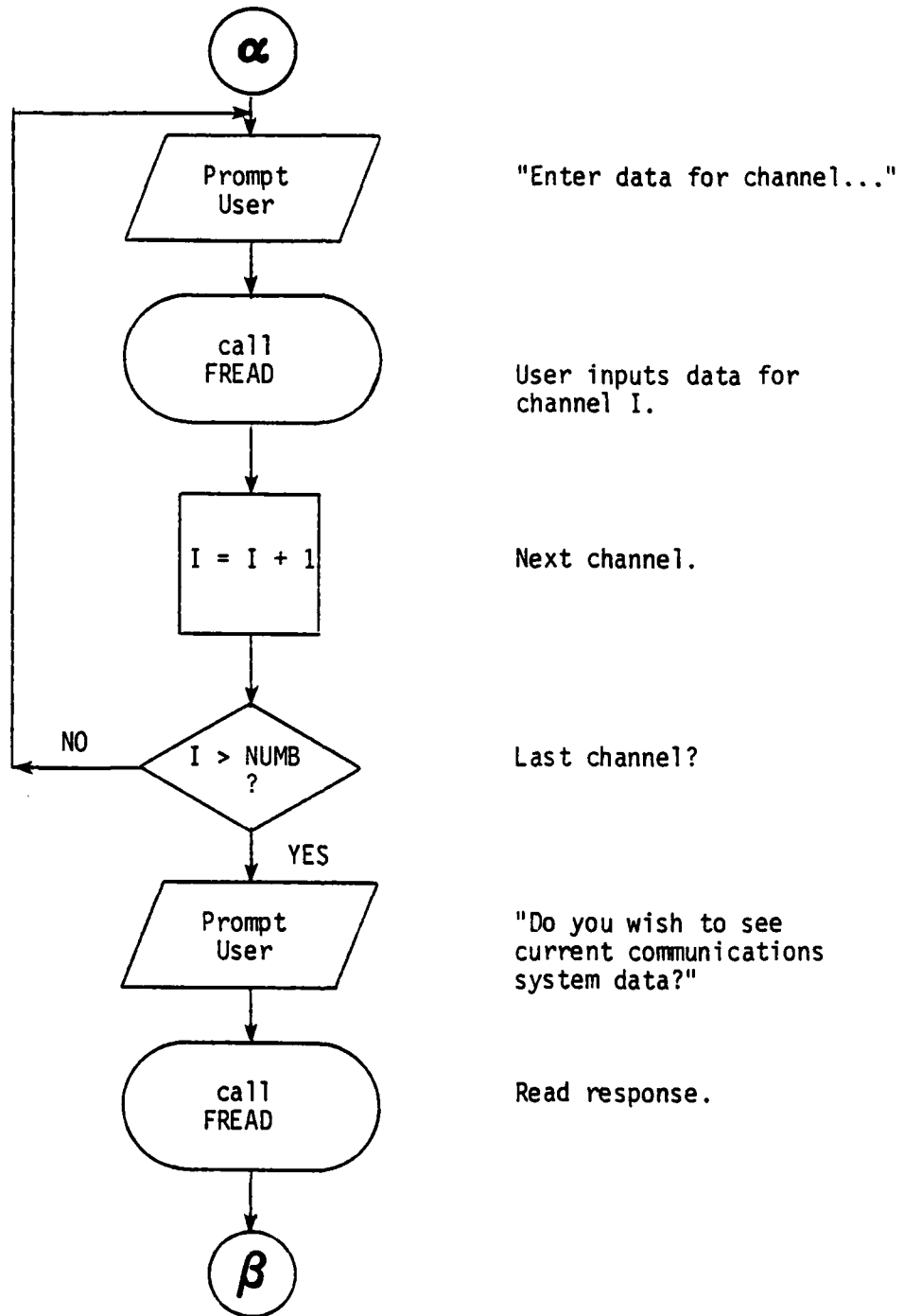
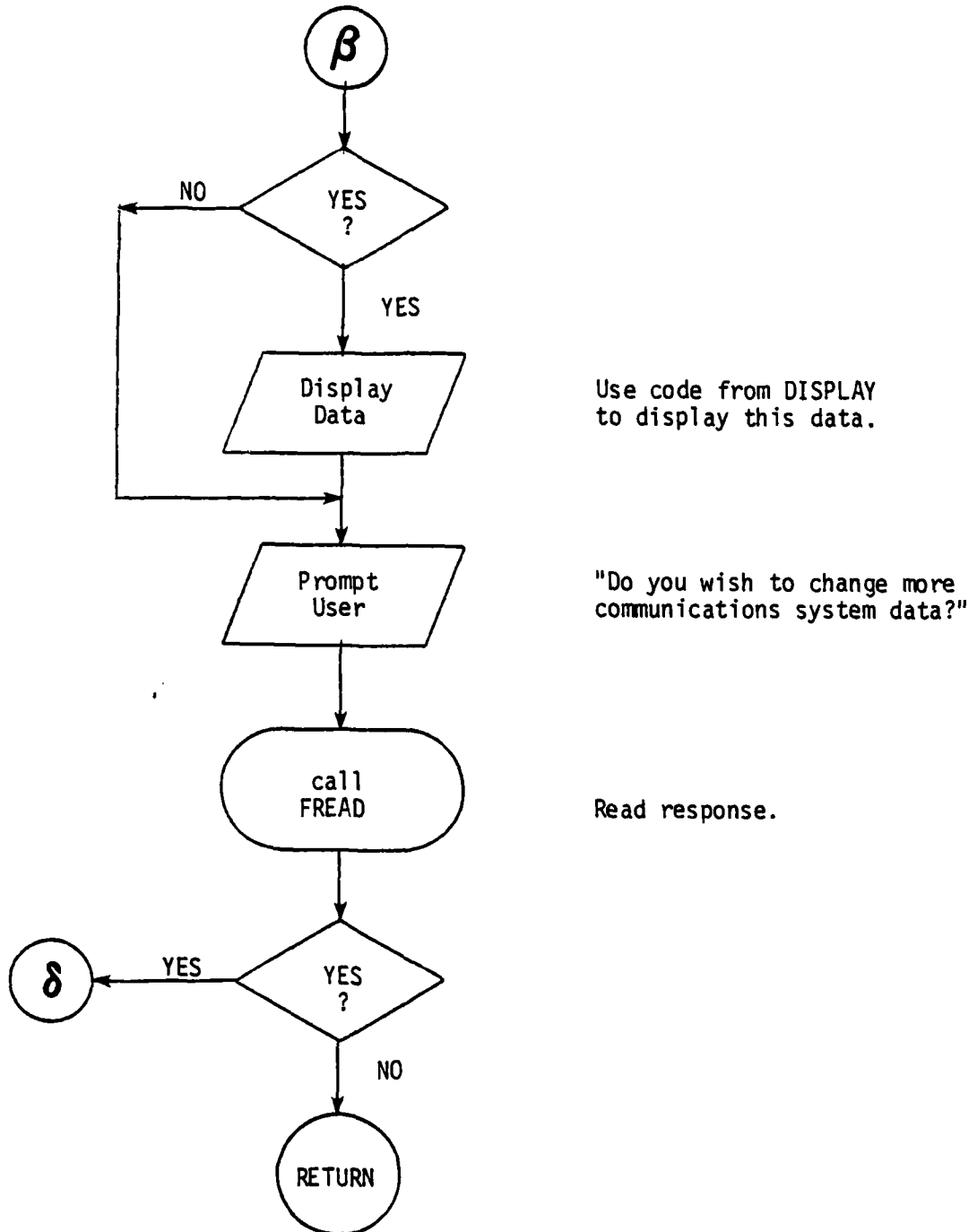
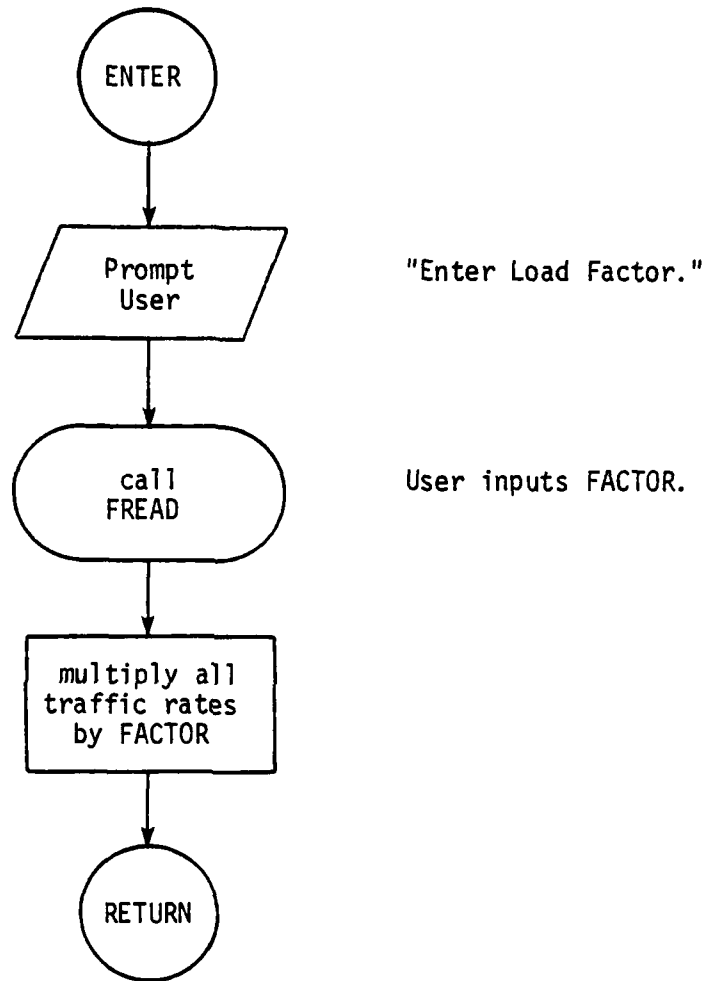


EXHIBIT 3-4: FLOWCHART OF SUBROUTINE COMSYS (Concluded)



## EXHIBIT 3-5: FLOWCHART OF SUBROUTINE LDFACT



## EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE

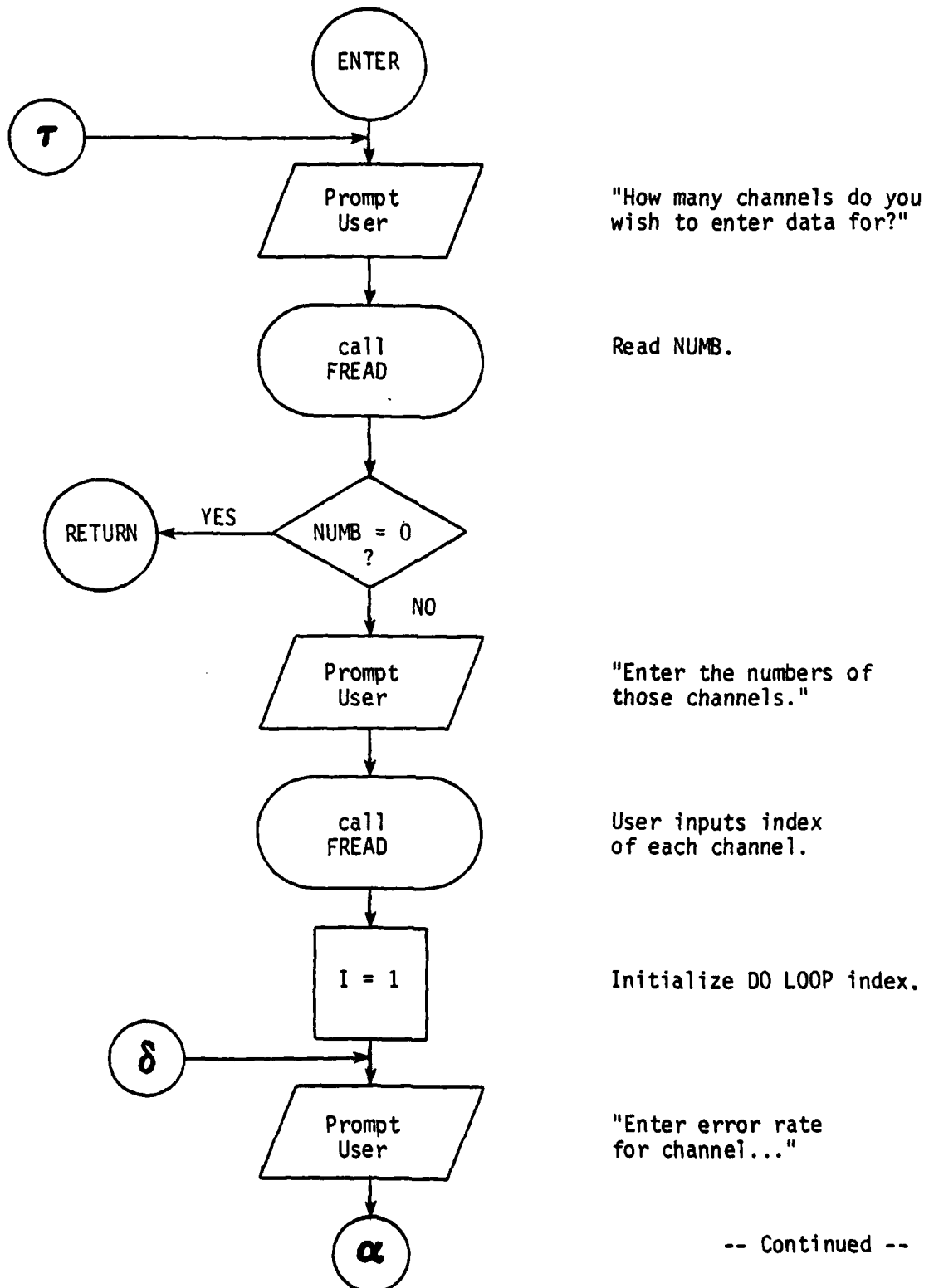


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE (Continued)

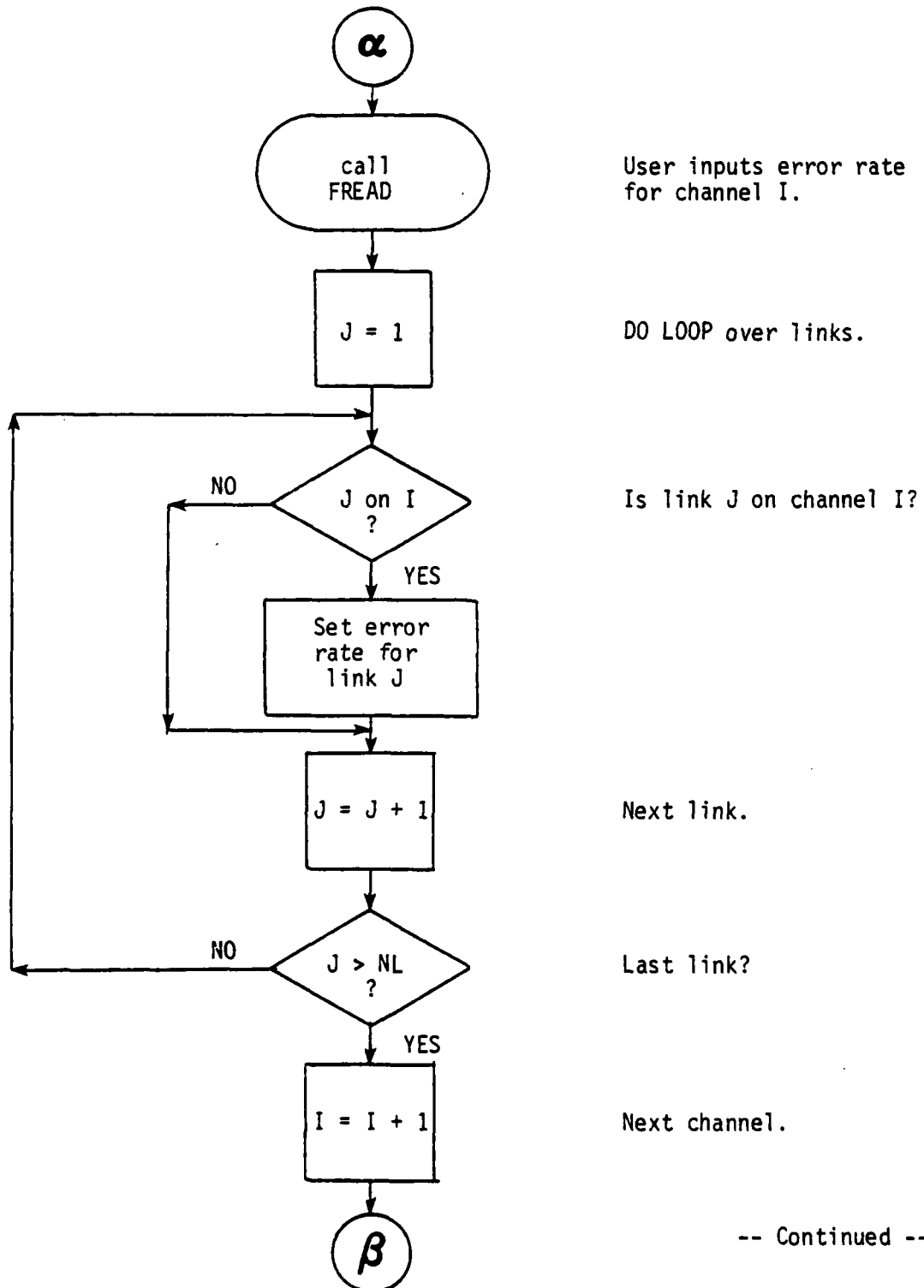


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE (Continued)

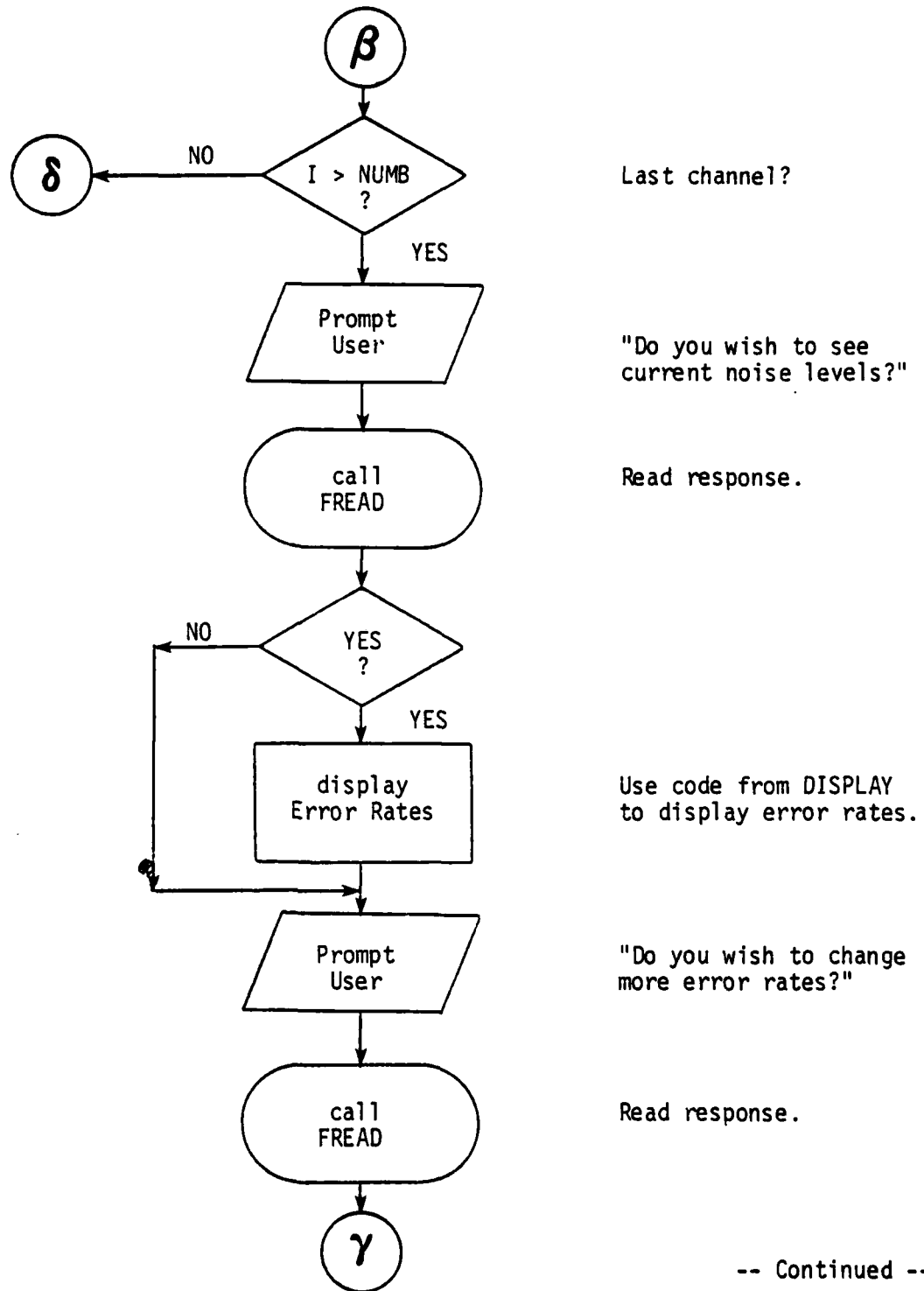
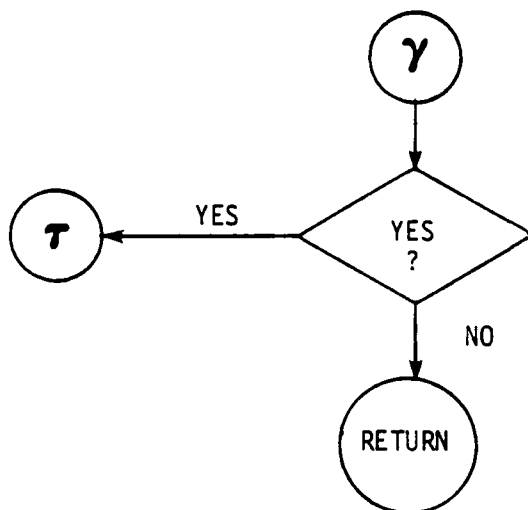


EXHIBIT 3-6: FLOWCHART OF SUBROUTINE NOISE (Concluded)



#### 4.0 PROGRAM DISPLAY

The third of the four programs in the TOS analysis package was written to assist the user in examining the contents of the data file. Program DISPLAY will read the data file and print the information with appropriate headings so that the user can easily examine the contents of the data file.

Program DISPLAY will mesh with the outputs of both program CREATE and program MODIFY. The read statements in program DISPLAY are copies of the write statements from the other two programs, with the word READ inserted in the place of WRITE. This was done to assure compatibility between the programs. Program DISPLAY is set up to operate on the Amdahl computer at The University of Michigan. This computer allows the user to specify input devices on the job control cards. Thus, the read statements in program DISPLAY are not the normal "READ(5, )" statements which are found in most FORTRAN codes, but rather are READ(2, ). Thus, the user is required to inform the operating system that device two contains the file which is to be displayed. If DISPLAY is implemented on another computer system, the compatibility with the read statements and the job control cards must be ensured so that program DISPLAY is linked to the proper data file.

Program DISPLAY was written in standard FORTRAN IV. The listing contained in Appendix A contains extensive comment lines which are used to identify the functions being performed. The first card read from the data file contains the counters which are used to control the DO LOOPS

which make up the remainder of program DISPLAY. The ordering of the route numbers, messages, and channels is the same as that contained in the other three programs and is used internally in DISPLAY. Program DISPLAY contains the option of reading data files which do not contain the Route Cross Message Array. If such a data file is used as the input for DISPLAY, the header for the Route Cross Message Array is printed in the output along with a message saying that the array is to be read in later.

## 5.0 PROGRAM COMPUTE

Program COMPUTE performs the computations specified by the model and displays the outputs. COMPUTE is organized into five modules, as shown in exhibit 5-1. Three of the program modules, the Traffic Flow Module, the Operating Statistics Module, and the Performance Measure Module, correspond to the three modules of the same names in the mathematical model.<sup>1</sup> The correspondence is not quite exact due to programming considerations such as producing efficient code and minimizing the program's memory requirements. Nonetheless, the primary content of the modules is unchanged.

The program documentation consists of a series of exhibits presenting: (1) flowcharts; (2) descriptions of the program variables; and (3) use of I/O units. Flowcharts for the Input and Initialization and Traffic Flow Modules are presented in exhibits 5-2 and 5-3. A flowchart for the Operating Statistics Module is presented in exhibit 5-4. In this flowchart, two segments are labeled A and B. More detailed flowcharts for these segments are presented in exhibits 5-5 and 5-6. Exhibit 5-7 contains a flowchart for a subroutine called by segment A of the Operating Statistics Module. Flowcharts for the Performance Measure Module and the Output Module are presented in exhibits 5-8 and 5-9. A list and description of the variables used in the program are presented in exhibit 5-10. Exhibit 5-11 shows the use of logical I/O units.

---

<sup>1</sup>See ARI Research Notes 80-13, Chapter 3.0.

## EXHIBIT 5-1: ORGANIZATION OF COMPUTE

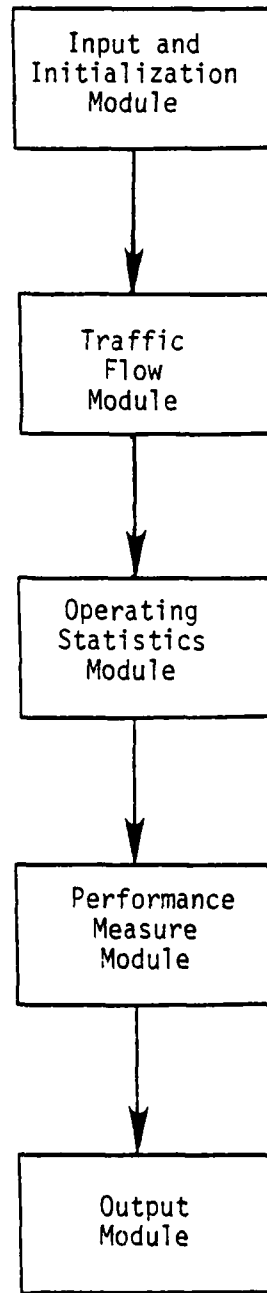


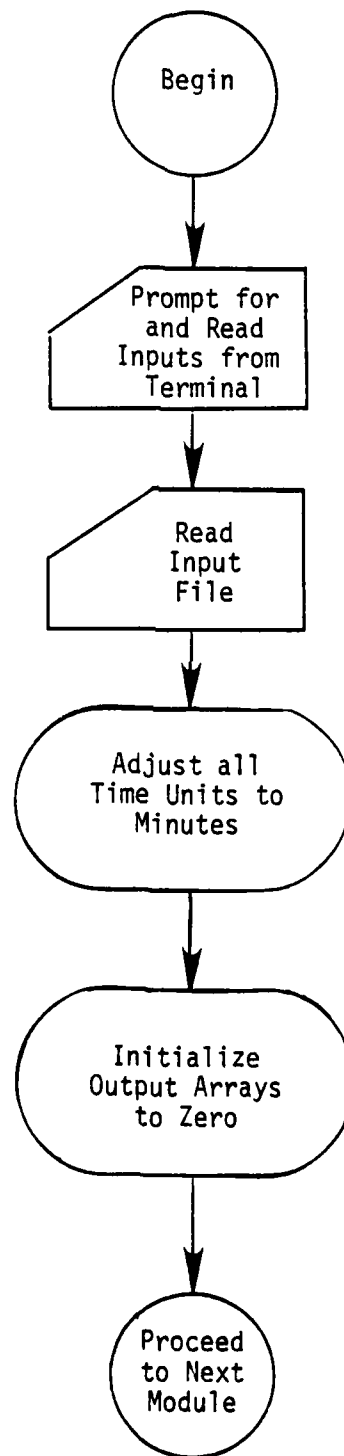
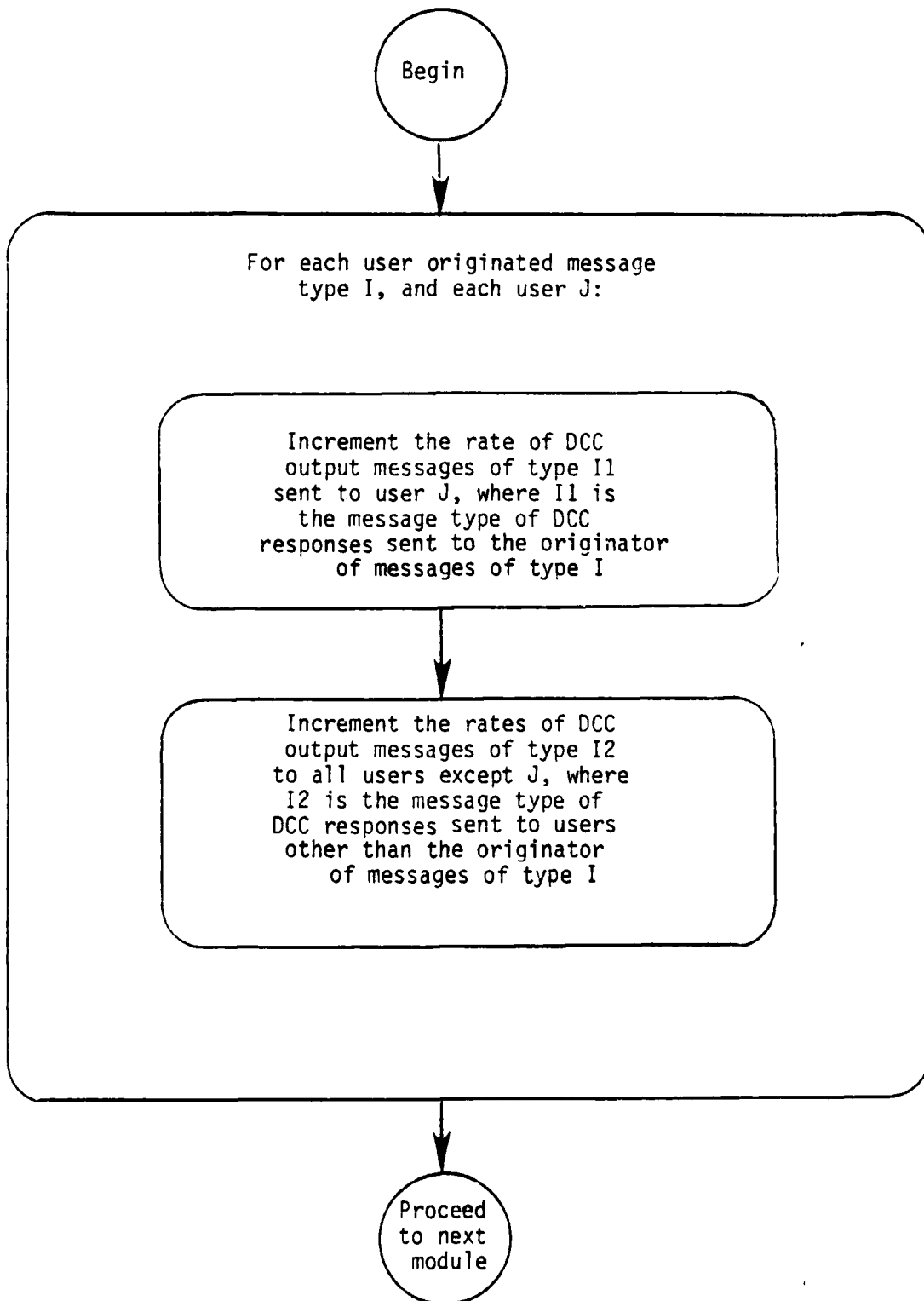
EXHIBIT 5-2: FLOWCHART FOR THE INPUT AND  
INITIALIZATION MODULE

EXHIBIT 5-3: FLOWCHART FOR THE TRAFFIC FLOW MODULE



## EXHIBIT 5-4: FLOWCHART FOR THE OPERATING STATISTICS MODULE

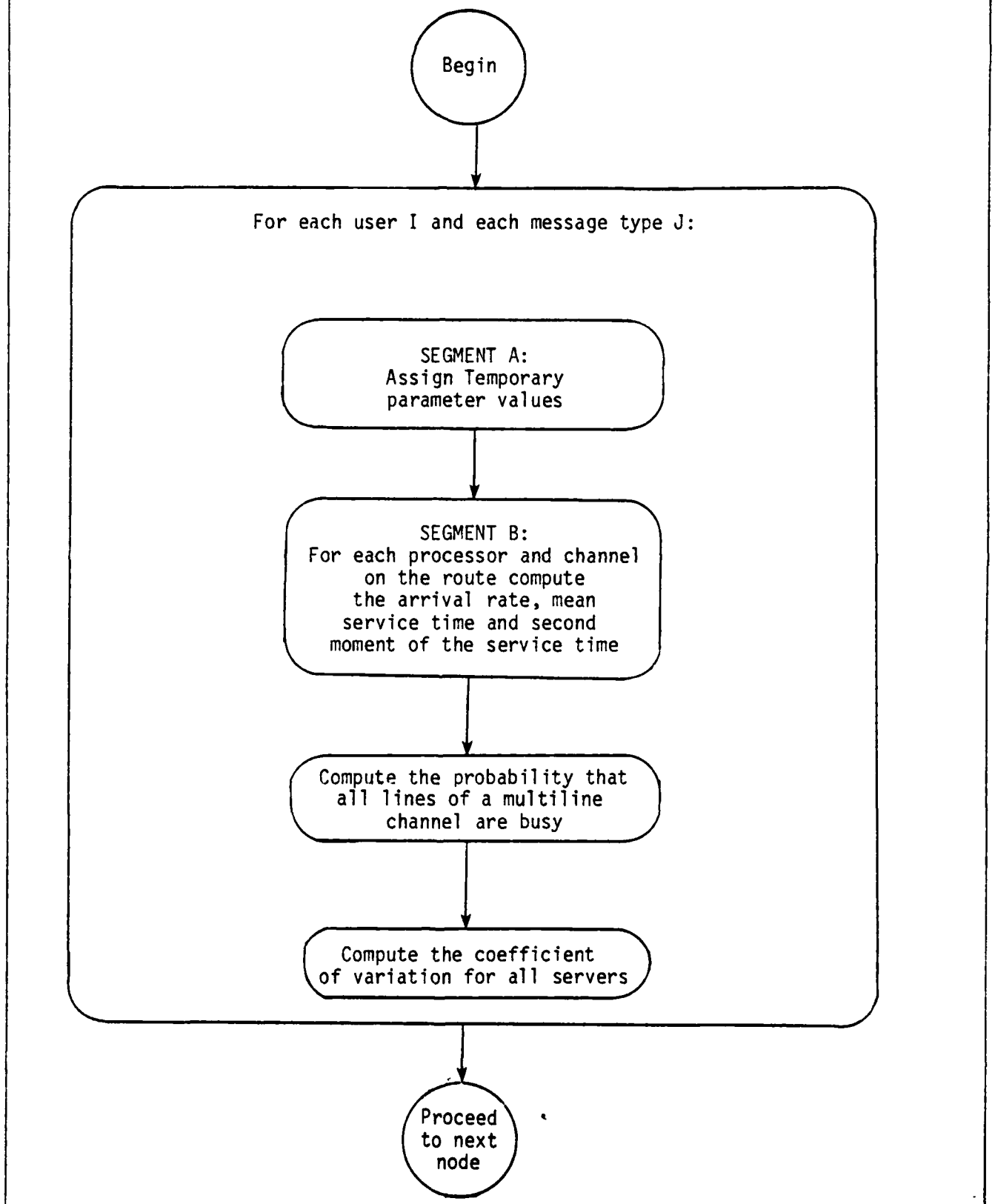


EXHIBIT 5-5: FLOWCHART FOR SEGMENT A OF THE OPERATING STATISTICS MODULE

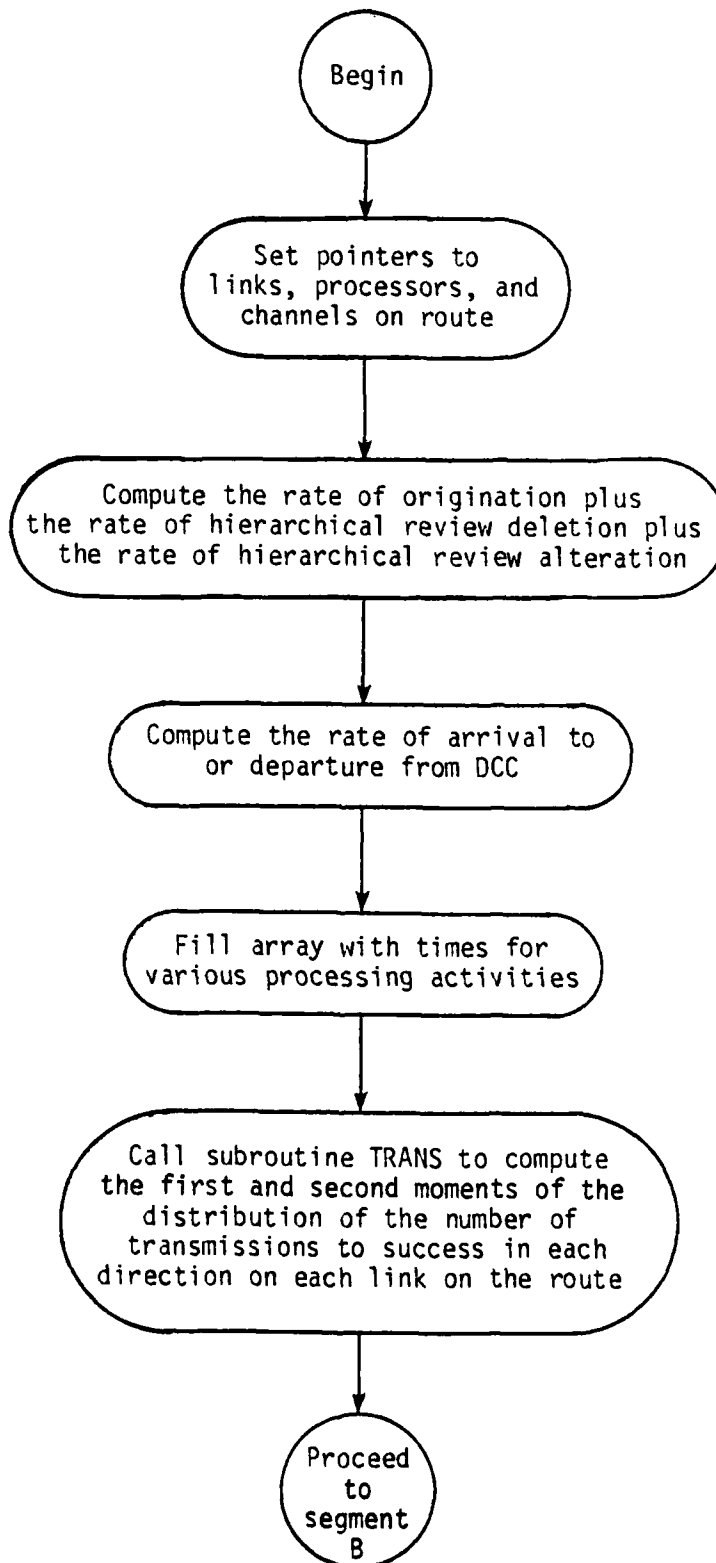


EXHIBIT 5-6: FLOWCHART FOR SEGMENT B OF OPERATING STATISTICS MODULE

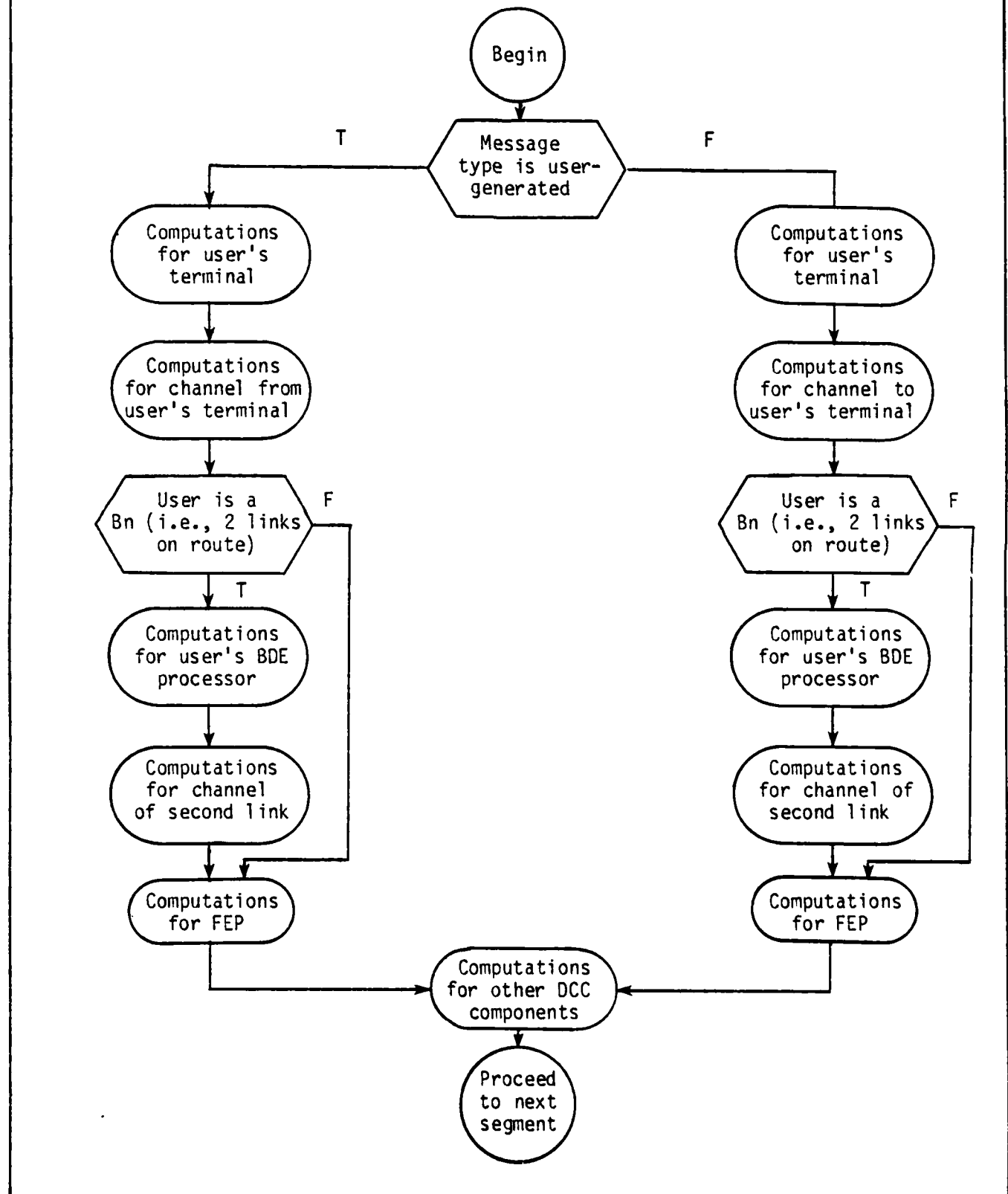
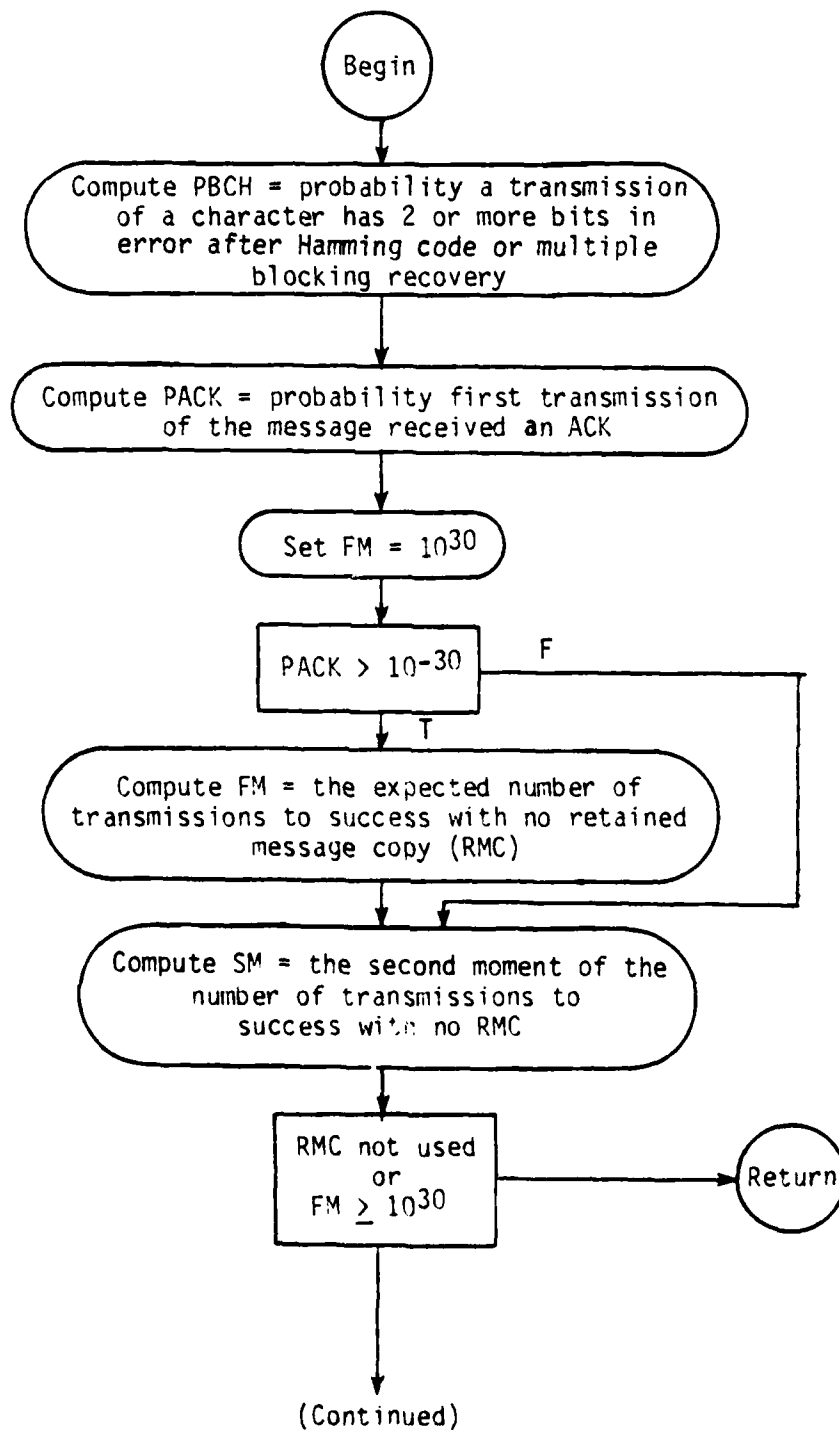
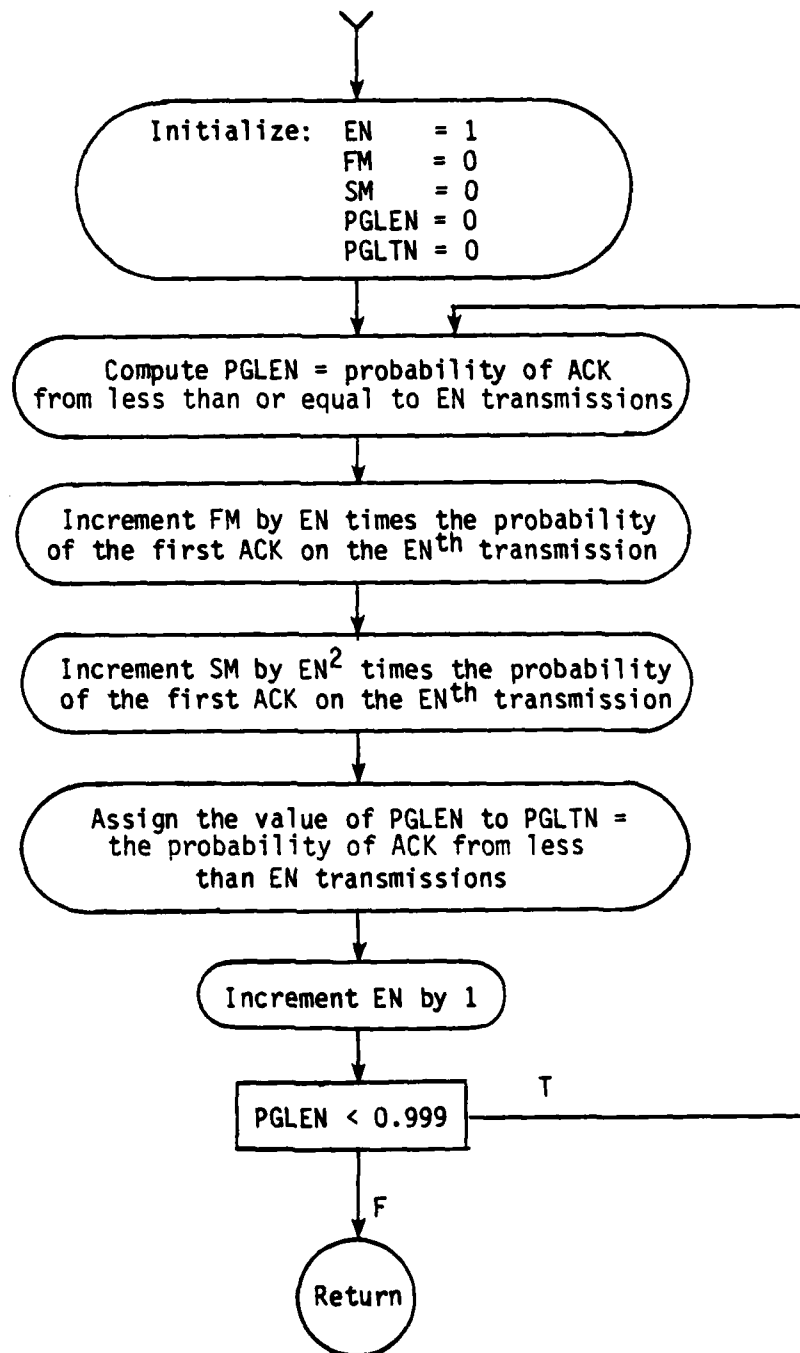


EXHIBIT 5-7: FLOWCHART FOR SUBROUTINE TRANS



## EXHIBIT 5-7: FLOWCHART FOR SUBROUTINE TRANS (Concluded)

(Continuation)



## EXHIBIT 5-8: FLOWCHART FOR THE PERFORMANCE MEASURE MODULE

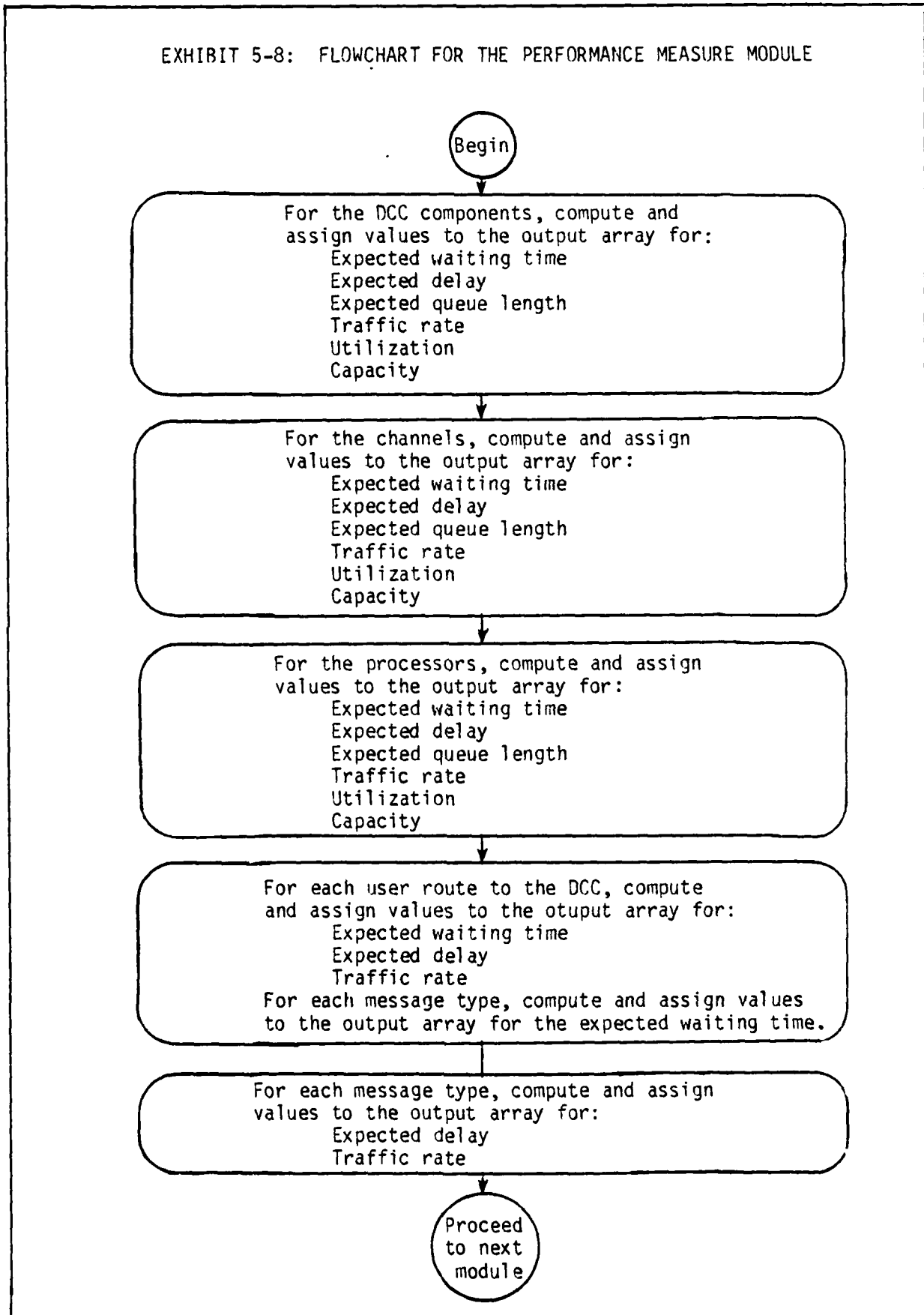
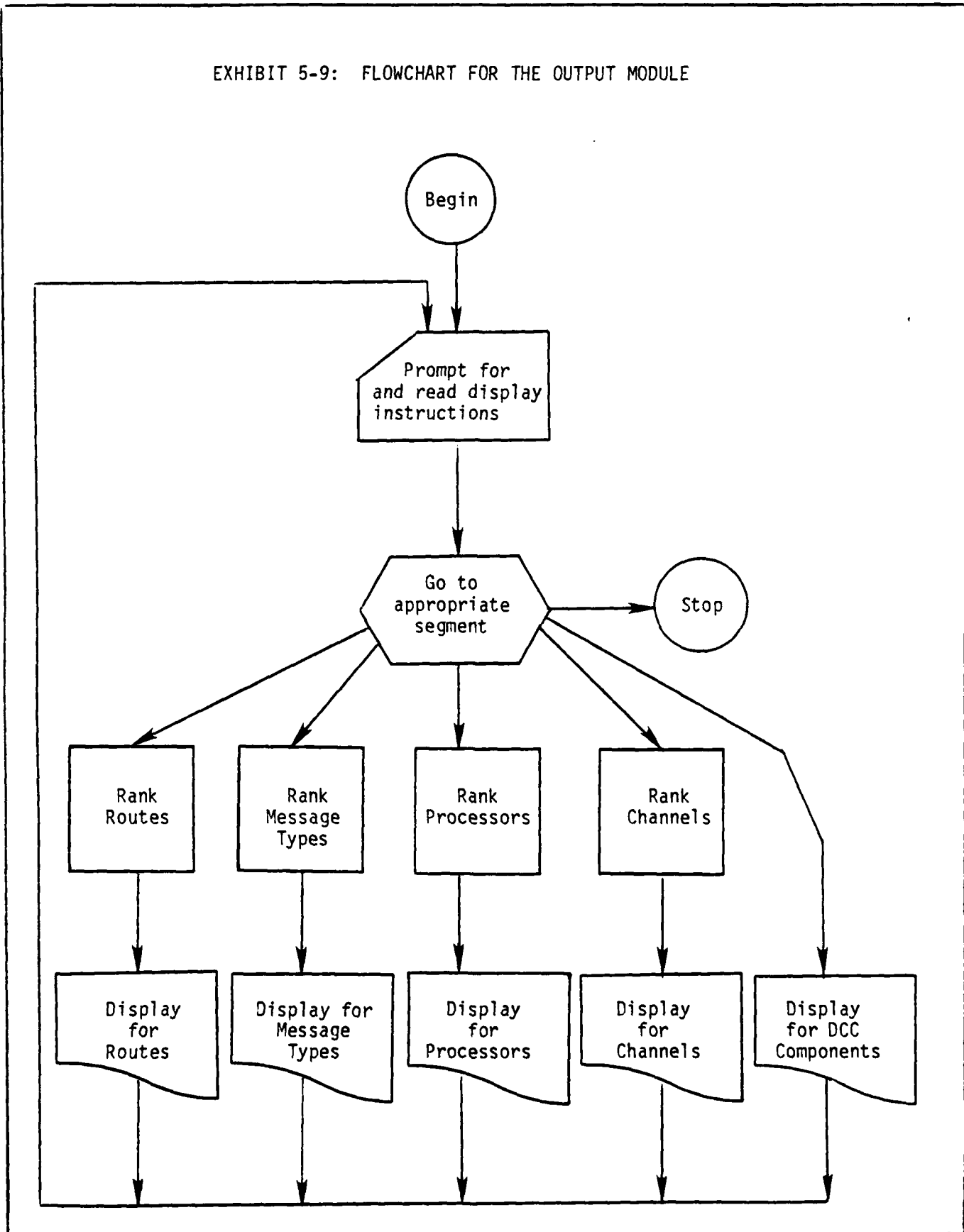


EXHIBIT 5-9: FLOWCHART FOR THE OUTPUT MODULE



## EXHIBIT 5-10: VARIABLES IN COMPUTE

<u>NAME</u>	<u>USE</u>
A	Temporary - Rate at which messages of type J travel to or from user I (including returns due to hierarchical review).
AM1	Temporary - Mean time for a processing or communications operation.
AM2	Temporary - Mean time for a processing or communications operation
AM3	Temporary - Mean time for a processing or communications operation.
B	Temporary - Rate at which messages of type J arrive or leave the DCC to/from user I.
CD (I, J)	Input - channel data: CD (I, 1) = overhead time per transmission on channel I CD (I, 2) = transmission rate on channel I CD (I, 3) = fraction of time used for voice on channel I CD (I, 4) = number of links on channel I
CNAM (I)	Input - name of channel I
COUT (I, J)	Output - channel outputs: COUT (I, 1) = expected delay on channel I COUT (I, 2) = expected queue length on channel I COUT (I, 3) = traffic rate on channel I COUT (I, 4) = utilization on channel I COUT (I, 5) = capacity on channel I COUT (I, 6) = rank on channel I
CS (I)	Temporary - probability all lines on channel I are busy.
C1	Temporary - pointer to channel of first link on a route.
C2	Temporary - pointer to channel of second link on a route.
DD (I)	Input - disk controller data: DD (1) = mean access time of message disk. DD (2) = mean access time of data disk.

-- Continued --

## EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
DOUT (I, J)	Output - DCC component (except FEP) output: D (I, 1) = expected delay D (I, 2) = expected queue length D (I, 3) = traffic rate D (I, 4) = utilization D (I, 5) = capacity I = 1    => DBP I = 2    => Message Disk Controller I = 3    => Data Disk Controller
FACT	Temporary - factorial
FM	Temporary - Mean number of transmissions to success in one direction on a link
FM1	Temporary - Mean number of transmissions to success in one direction on a link
FM2	Temporary - Mean number of transmissions to success in one direction on a link
FM3	Temporary - Mean number of transmissions to success in one direction on a link
FM4	Temporary - Mean number of transmissions to success in one direction on a link
GPD (I, J)	Input - Processing time information: GPD (1, J) = TCS origination time GPD (2, J) = TCS send time GPD (3, J) = TCS receive time GPD (5, J) = TCS terminate time GPD (6, J) = FEP terminate time GPD (7, J) = TCT terminate time GPD (8, J) = TCT send time GPD (9, J) = TCT receive time GPD (10, J) = TCT terminate time GPD (11, J) = FEP originate time J = 1 => per message J = 2 => per character
I	Temporary - Counter or pointer
IA	Input - Pointer to current output table

-- Continued --

## EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
IB	Input - Pointer to column for ranking.
IC	Input - Reordering switch.
IP	Temporary - Counter.
IPD (I)	Input - Processor data - type of processor at node I, 1 => TCS, 2 => TCT.
IRMC	Input - Retained Message Copy (RMC) Switch.
I1	Temporary - Pointer to message type of outputs in response to inputs of type I sent to the originator of the message of type I.
I2	Temporary - Pointer to message type of outputs in response to inputs of type I sent to users other than the originator of the message of type I.
J	Temporary - Counter and pointer.
JPD	Temporary - Type of current processor.
K	Temporary - Counter and pointer.
LARGE	Temporary - $10^{40}$ .
LD (I,J)	Input - Link data on error rates LD (I, 1) = Error rate on Link I for messages sent towards the DCC. LD (I, 2) = Error rate on Link I for messages sent towards a user.
LP (I,J)	Input - Pointers to devices on links. LP (I, 1) = Processor closest to user on Link I LP (I, 2) = Channel used by Link I LP (I, 3) = Processor closest to DCC on Link I
L1	Temporary - Pointer to the first link on the route from a user to the DCC.
L2	Temporary - Pointer to the second link on the route from a user to the DCC. Value zero if there is no second link.

-- Continued --

## EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
MD (I,J)	Input - Message data: MD (I, 1) = Number of characters in Message Type I MD (I, 2) = DBP processing time of Message Type I MD (I, 3) = Number of message disk accesses for messages of type I MD (I, 4) = Number of Data Disk accesses for messages of type I
MDJ1	Temporary - Length of current message type
MNAM (I)	Input - Message Names
MOUT (I, J)	Output - Message outputs: MOUT (I, 2) = Expected delay in Message Type I MOUT (I, 2) = Traffic rate in Message Type I MOUT (I, 3) = Rank in Message Type I
MP (I,J)	Input - MP (I, 1) = Message type of same-route outputs in response to messages of type I MP (I, 2) = Message type of other-route outputs in response to messages of type I
MS	Temporary.
NBLOCK	Input - Blocking number
NC	Input - Number of channels
NL	Input - Number of links
NM	Input - Number of message types.
NN	Input - Number of nodes
NO (I, J)	Temporary - NO (I, 1) = Accumulator for Node I mean service time NO (I, 2) = Node I arrival rate. NO (I, 3) = Accumulator for Node I standard deviation of service time.
NON1	Temporary - Utilization of a component
NP	Input - Number of processors

-- Continued --

## EXHIBIT 5-10: VARIABLES IN COMPUTE (Continued)

<u>NAME</u>	<u>USE</u>
NP11	Temporary - NP-1.
NR	Input - Number of users.
NUM	Input - Number of user input message types.
N1	Temporary - General purpose pointer.
N2	Temporary - General purpose pointer.
N3	Temporary - General purpose pointer.
N4	Temporary - General purpose pointer.
N5	Temporary - General purpose pointer.
PD (I)	Input - Number of templates for screening messages at a processor I. (0 for TCTs).
POUT (I, J)	Output - Processor outputs: POUT (I, 1) = expected delay at processor I POUT (I, 2) = expected queue length at processor I POUT (I, 3) = traffic rate at processor I POUT (I, 4) = utilization at processor I POUT (I, 5) = capacity at processor I POUT (I, 6) = rank at processor I
RM (I, J, K)	Input - Route cross message type data: RM (I, J, 1) = Initiation rate for user generated messages, proportionality constant for TOS-generated messages. RM (I, J, 2) = Second proportionality constant for TOS-generated messages. RM (I, J, 3) = Probability of hierarchical review deletion for battalion user-generated messages, else zero. RM (I, J, 4) = Probability of hierarchical review alteration for battalion-generated messages, else zero.
RMO (I, J)	Temporary - Rate of each type of message at each node
RNAM (I)	Input - Names of users.

-- Continued --

## EXHIBIT 5-10: VARIABLES IN COMPUTE (Concluded)

<u>NAME</u>	<u>USE</u>
ROUT (I, J)	Output - Route outputs: ROUT (I, 1) = Expected delay on route I ROUT (I, 2) = Traffic rate on route I ROUT (I, 3) = Rank of route I
RP (I, J)	Input - Pointers to the first and second links on route I. (Second pointer zero if no second link.)
SM	Temporary - Second moment of the number of transmissions to success on a link to the FEP.
SM1	Temporary - Second moment of the number of transmissions to success on a link.
SM2	Temporary - Second moment of the number of transmissions to success on a link.
SM3	Temporary - Second moment of the number of transmissions to success on a link.
SM4	Temporary - Second moment of the number of transmissions to success on a link.
SUM	Temporary - Scratch pad.
T (I, J, K)	Temporary - Times for various processing activities.
VT	Temporary - Scratch pad.
VM1	Temporary - Second moment of service time.
VM2	Temporary - Second moment of service time.
VM3	Temporary - Second moment of service time.
X	Temporary - Scratch pad.

## EXHIBIT 5-11: LOGICAL I/O UNIT ASSIGNMENTS

<u>UNIT</u>	<u>USE</u>
5	Write to operator at terminal.
6	Read from operator at terminal.
7	Read data file.
8	Write output tables.

## APPENDIX A: LISTINGS OF TOS SIMULATION PACKAGE PROGRAMS

This appendix contains source listings of the programs which make up the TOS Simulation Package. These programs are named CREATE, DISPLAY, MODIFY, and COMPUTE. The statements within a single routine can be referenced either by a line number or by an internal statement number. Line numbers, shown on the left-hand side of each listing, begin with unity for each program and continue sequentially to the end of the program, each line being assigned a number one integer greater than its predecessor. Internal statement numbers, appearing on the right-hand side of each listing, begin with unit for each routine within a program. Unlike line numbers, internal statement numbers index FORTRAN statements. Thus, there are no internal statement numbers assigned to comment lines, and just one integer assigned to multiline FORTRAN statements.



&lt;PAGE 2&gt;

```

1 DATE:02-05-80,10:40 CWRN:IDSH FILE:CREATE
1

```

LSN

```

59 C      PATTALIONS
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C

45 CONTINUE

DO 60 I = 1, NPEE
  IF (NON(I) .EQ. (J)) GO TO 60
  III = NON(I)

DO 55 J = 1, III
  WRITE (6,50) J, RENAM(I)
  FORMAT (' ENTER NAME OF BATTALION ', I1, ' ,PRIGADE ',
1      A8)
  CALL FREAD(5, 'S:', UNAME(I,J), 8)
  WRITE (6,30) UNAME(I,J)
  CALL FREAD(5, 'S:', UN(I,J), 4)
  CALL CHECK(UN(I,J))
  IF (IV .EQ. 1) GO TO 160
55 CONTINUE

60 CONTINUE

C      OTHER TOS USERS

65 CONTINUE
70 FORMAT (' ENTER NUMBER OF OTHER TOS USERS')
CALL FREAD(5, 'I:', NCH)
IF (NCH .EQ. (0)) GO TO 85

DO 80 I = 1, NCH
  WRITE (6,75) I
  FORMAT (' ENTER NAME OF OTHER TOS USER ', I2)
  CALL FREAD(5, 'S:', CUNAME(I), 9)
  WRITE (6,10) CUNAME(I)
  CALL FREAD(5, 'S:', OU(I), 4)
  CALL CHECK(CU(I))
  IF (IV .EQ. 1 .AND. NCH .GT. 1) GO TO 180
80 CONTINUE

C      COMMUNICATIONS NETWORK

85 CONTINUE
90 FORMAT (' HOW MANY COMMUNICATIONS CHANNELS ARE THERE?')
CALL FREAD(5, 'I:', NC)

C      NAMES OF CHANNELS

DO 100 I = 1, NC
  WRITE (6,95) I
  FORMAT (' ENTER NAME OF CHANNEL ', I2)
  CALL FREAD(5, 'S:', CHAN(I), 8)
  IF (IV .EQ. 1 .AND. NC .GT. 1) GO TO 195
  IF (IV .EQ. 1 .AND. NC .EQ. 1) GO TO 210
100 CONTINUE

C      BATTALION - BRIGADE

```

&lt;PAGE 2&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:CREATE /////

&lt;PAGE 2&gt;

LINE	CODE	TEXT	DATE	TIME	USER	FILE	STATUS
117	C	IF (NDEF -FO. 0-) GO TO 130	02-05-80	14:40	CHIEF:IDSH	FILE:CRFATF	
118							
119	C	DO 115 I = 1, NDEF					
120		IF (NDEF(I) -FO. 0) GO TO 115					
121		111 = NDEF(I)					
122	C						
123		PO 110 J = 1, 111					
124		WRITE (6,105) BNAME(I,J), UNAME(I)					
125	105	FORMAT (' ENTER NAME OF CHANNEL CONNECTING BATTALION ',					
126		AB, ' WITH BRIGADE ', AB)					
127		CALL FREAD(5, 'S:', NINCK(I,J), 0)					
128		CALL CHECK(NINCK(I,J), CHAM, NC, LINK(I,J))					
129	110	CONTINUE					
130	C						
131	C	115 CONTINUE					
132	C						
133	C	BRIGADE - FEP					
134	C						
135		PO 125 I = 1, NDEF					
136		WRITE (6,120) EDNAME(I)					
137	120	FORMAT (' ENTER NAME OF CHANNEL CONNECTING BRIGADE ', AB,					
138		I, AND THE FEP')					
139		CALL FREAD(5, 'S:', NINCK(I), 0)					
140		CALL CHECK(NINCK(I), CHAM, NC, LINK(I))					
141	125	CONTINUE					
142	C						
143	C	OTHER TOS USERS - FEP					
144	C						
145	C	130 CONTINUE					
146		IF (MOU -FO. 0) GO TO 250					
147	C						
148		DO 140 I = 1, MOU					
149		WRITE (6,135) CNAME(I)					
150	135	FORMAT (' ENTER NAME OF CHANNEL CONNECTING ', AR,					
151		I, AND THE FEP')					
152		CALL FREAD(5, 'S:', NINCK(I), 0)					
153		CALL CHECK(NINCK(I), CHAM, NC, LINK(I))					
154	140	CONTINUE					
155	C						
156		GO TO 255					
157	C						
158	C	SHORT PROMPTS FOR BRIGADE INFORMATION					
159	C						
160	C	145 CONTINUE					
161	C						
162		DO 155 I = 2, NDEF					
163		WRITE (6,150) I					
164	150	FORMAT (' NAME, TYPE PROCESSOR, # OF RM IN BDE ', I2)					
165		CALL FREAD(5, 'S:', BNAME(I), 0, 'S:', 0, 'I:',					
166		I, MEN(I))					
167		CALL CHECK(BD(I))					
168	155	CONTINUE					
169	C						
170		GO TO 45					
171	C						
172	C	SHORT PROMPTS FOR BATTALION INFORMATION					
173	C						
174	C	160 CONTINUE					

LINE	CODE	TEXT	ISN
175		JJ = 2	104
176		II = 1	105
177	C		106
178		DO 175 I = II, NUDE	107
179		IF (JJ .GT. NNN(I)) GO TO 175	108
180		IP (NNN(I) .EQ. 0) GO TO 175	109
181		IF (I .NE. II) JJ = 1	110
182		III = NNN(I)	111
183	C		112
184		GO 170 J = JJ, III	113
185		WRITE (6,165) J, BNAME(I)	114
186		FORMAT (' NAME, PROCESSOR TYPE, PM, ', I1, ', ', A8)	115
187		CALL FREAD(5, 'S:', BNAME(I), 8, 'S:', BN(I), 4)	116
188		CALL CHECKON(I,J)	117
189		CONTINUE	118
190	C		119
191		CONTINUE	120
192	C		121
193		CONTINUE	122
194		GO TO 65	123
195	C		124
196		SHORT PROMPTS FOR OTHER LOS USERS INFORMATION	125
197	C		126
198		CONTINUE	127
199	C		128
200		DO 190 J = 2, NDU	129
201		WRITE (6,185)	130
202		FORMAT (' NAME, PROCESSOR TYPE')	131
203		CALL FREAD(5, 'S:', CNAME(I), 8, 'S:', ON(I), 4)	132
204		CALL CHECK(ON(I))	133
205		CONTINUE	134
206	C		135
207		GO TO 85	136
208	C		137
209		SHORT PROMPTS FOR CHANNEL INFORMATION	138
210	C		139
211		CHANNEL NAMES	140
212	C		141
213		CONTINUE	142
214	C		143
215		DO 205 I = 2, NC	144
216		WRITE (6,200) I	145
217		FORMAT (' CHANNEL', 12, '')	146
218		CALL FREAD(5, 'S:', CNAME(I), 8)	147
219		CONTINUE	148
220	C		149
221		BRIGADE - FFP	150
222	C		151
223		CONTINUE	152
224		IF (NDE .EQ. 0) GO TO 245	153
225		WRITE (6,210) PNAME(I)	154
226		CALL FREAD(5, 'S:', NLINC(I), 4)	155
227		CALL CHECK(INC(I), CNAME, NC, LINC(I))	156
228		IF (NDE .EQ. 1) GO TO 225	157
229	C		158
230		DO 220 I = 2, NUDE	159
231		WRITE (6,215) ENAME(I)	160
232		FORMAT (' ', AP, ' TO FFP')	161
233		CALL FREAD(5, 'S:', NLINC(I), 4)	162

LINE	CODE	TEXT	DATE:02-05-80,18:40	USER:IPSH	FILE:CREATE
233		CALL CHECK1(INLNK(I), CNAME, NC, LINK(I))			
234		220 CONTINUE			
235	C				
236		ATTENTION - BRIGADE			
237	C				
238		225 CONTINUE			
239	C				
240		DO 240 I = 1, NDEE			
241		IF (NDN(I) .EQ. 0) GO TO 240			
242		JI = NDN(I)			
243	C				
244		DO 245 J = 1, JI			
245		WRITE (6,230) BNAMEP(I,J), DNAMEP(I)			
246		FORMAT (' ', A8, 'TO ', A9)			
247		CALL FREAL(5, 'S:', MLNK(I,J), R)			
248		CALL CHECK1(MLNK(I,J), CNAME, NC, LINK(I,J))			
249		245 CONTINUE			
250	C				
251		240 CONTINUE			
252	C				
253		OTHER TOS USERS - PEP			
254	C				
255		245 CONTINUE			
256		IF (NCN .EQ. 0) GO TO 255			
257	C				
258		DO 255 I = 1, NOH			
259		IF (NDPE .EQ. 0 .AND. I .EQ. 1) WRITE (6,115) OUNAME(I)			
260		IF (NDPE .EQ. 0 .AND. I .EQ. 1) CALL FREAL(5, 'S:',			
261		1 MLNK(I), R)			
262		IF (NDPE .EQ. 0 .AND. I .EQ. 1) GO TO 250			
263		WRITE (6,215) CUNAME(I)			
264		CALL FREAL(5, 'S:', MLNK(I), R)			
265		CALL CHECK1(MLNK(I), CNAME, NC, LINK(I))			
266		255 CONTINUE			
267	C				
268		***** MAKE CONFIGURATION ASSIGNMENTS *****			
269	C				
270		COUNT NUMBER OF ROUTES			
271	C				
272		260 CONTINUE			
273		NR = NDPE + NOH			
274		IF (NDPE .EQ. 0) GO TO 270			
275	C				
276		DO 265 I = 1, NDEE			
277		NR = NR + NDN(I)			
278		RL = NR			
279		265 CONTINUE			
280	C				
281		270 CONTINUE			
282		NP = NR + 1			
283		NN = PD + NC + 1			
284	C				
285		FORTE AND LINK PCINTERS, OTHER TOS USERS			
286	C				
287		IF (NCN .EQ. 0) GO TO 290			
288		DO 275 I = 1, NOH			
289		BNAME(I) = OUNAME(I)			
290	C				

<PAGE 5>

>>>> MAIN PROGRAM <<<<<

>>>> FILE:CREATE <<<<<



```

C
340 ITCT = 0
350 DO 310 I = 1, NUM
360 IF (IPD(I) -EQ. 1) ITCS = 1
370 IF (IPD(I) -EQ. 2) ITCT = 2
380 IF (ITCT + ITCS .EQ. 3) GO TO 315
390 CONTINUE
400 CONTINUE
410 ML = NUM
420 ***** MESSAGE DATA *****
430 WRITE (6,320)
440 FORMAT (' ENTER NUMBER OF MSG TYPES AND NUMBER OF USER ',
450 ' INPUT MSG TYPES')
460 CALL FREAD(-1, 'DELI')
470 CALL FPAE(5, '21:', NO, NUM)
480 CALL FREAD(-2, 'DELI', '//1/')
490 ***** USER INITIATED MESSAGE NAMES *****
500 IP (NPM .PO. 0) GO TC 115
510 DO 330 I = 1, NUM
520 WRITE (6,325) I
530 FORMAT (' ENTER NAME OF USER MSG TYPE ', I2)
540 CALL FPAE(5, 'S:', MMAM(I), 8)
550 IF (IV .FO. 1 .AND. NM .GT. 1) GO TO 385
560 IF (IV .FO. 1 .AND. NM .EQ. 1 .AND. NM .GT. NM)
570 GO TO 400
580 CONTINUE
590 TOS GENERATED MESSAGE NAMES
600 IF (NUM .PO. NP) GO TO 350
610 JJ = NUM + 1
620 DO 340 I = JJ, NM
630 JJ = I - NUM
640 WRITE (6,340) JJ
650 FORMAT (' ENTER NAME OF TOS MSG TYPE ', I2)
660 CALL FPAE(5, 'S:', MMAM(I), 8)
670 IF (IV .FO. 1 .AND. NM .EQ. 0 .AND. NM .GT. 1)
680 GO TO 410
690 CONTINUE
700 ***** USER MESSAGE DATA *****
710 PEAN LENGTHS
720 CONTINUE
730 CALL FREAD(-1, 'DELI')
740 IF (NM .PO. 0) GO TO 375
750 DO 370 I = 1, NUM
760 WRITE (6,355) PMAM(I)

```

<PAGE 7>

>>>> MAIN PROGRAM <<<<

//// PIVE:CHRATE ////

PAGE 18

<PAGE 8>

DATE: 02-05-80, 14:40 OWNER: LOSM FILE: CREATE

<PAGE 9>

```

258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224

```

<PAGE 8>

```
>>>> MAIN PROGRAM <<<<
```

FILE:CREATP ////

SPACE 82

```

465 CALL FPRAD(5, 'S:', MNAME(1), 0)
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

C 315
C 316
C 317
C 318
C 319
C 320
C 321
C 322
C 323
C 324
C 325
C 326
C 327
C 328
C 329
C 330
C 331
C 332
C 333
C 334
C 335
C 336
C 337
C 338
C 339
C 340
C 341
C 342
C 343
C 344
C 345
C 346
C 347
C 348
C 349
C 350
C 351
C 352
C 353
C 354
C 355
C 356
C 357
C 358
C 359
C 360
C 361
C 362
C 363
C 364
C 365

C 400 J = 1, NH
      IF (J.GI. NH) GO TO 400
C 401
C 402
C 403
C 404
C 405
C 406
C 407
C 408
C 409
C 410
C 411
C 412
C 413
C 414
C 415
C 416
C 417
C 418
C 419
C 420
C 421
C 422
C 423
C 424
C 425
C 426
C 427
C 428
C 429
C 430
C 431
C 432
C 433
C 434
C 435
C 436
C 437
C 438
C 439
C 440
C 441
C 442
C 443
C 444
C 445
C 446
C 447
C 448
C 449
C 450
C 451
C 452
C 453
C 454
C 455
C 456
C 457
C 458
C 459
C 460
C 461
C 462
C 463
C 464
C 465
C 466
C 467
C 468
C 469
C 470
C 471
C 472
C 473
C 474
C 475
C 476
C 477
C 478
C 479
C 480
C 481
C 482
C 483
C 484
C 485
C 486
C 487
C 488
C 489
C 490
C 491
C 492
C 493
C 494
C 495
C 496
C 497
C 498
C 499
C 500
C 501
C 502
C 503
C 504
C 505
C 506
C 507
C 508
C 509
C 510
C 511
C 512
C 513
C 514
C 515
C 516
C 517
C 518
C 519
C 520
C 521
C 522
C 523
C 524
C 525
C 526
C 527
C 528
C 529
C 530
C 531
C 532
C 533
C 534
C 535
C 536
C 537
C 538
C 539
C 540
C 541
C 542
C 543
C 544
C 545
C 546
C 547
C 548
C 549
C 550
C 551
C 552
C 553
C 554
C 555
C 556
C 557
C 558
C 559
C 560
C 561
C 562
C 563
C 564
C 565
C 566
C 567
C 568
C 569
C 570
C 571
C 572
C 573
C 574
C 575
C 576
C 577
C 578
C 579
C 580
C 581
C 582
C 583
C 584
C 585
C 586
C 587
C 588
C 589
C 590
C 591
C 592
C 593
C 594
C 595
C 596
C 597
C 598
C 599
C 600
C 601
C 602
C 603
C 604
C 605
C 606
C 607
C 608
C 609
C 610
C 611
C 612
C 613
C 614
C 615
C 616
C 617
C 618
C 619
C 620
C 621
C 622
C 623
C 624
C 625
C 626
C 627
C 628
C 629
C 630
C 631
C 632
C 633
C 634
C 635
C 636
C 637
C 638
C 639
C 640
C 641
C 642
C 643
C 644
C 645
C 646
C 647
C 648
C 649
C 650
C 651
C 652
C 653
C 654
C 655
C 656
C 657
C 658
C 659
C 660
C 661
C 662
C 663
C 664
C 665
C 666
C 667
C 668
C 669
C 670
C 671
C 672
C 673
C 674
C 675
C 676
C 677
C 678
C 679
C 680
C 681
C 682
C 683
C 684
C 685
C 686
C 687
C 688
C 689
C 690
C 691
C 692
C 693
C 694
C 695
C 696
C 697
C 698
C 699
C 700
C 701
C 702
C 703
C 704
C 705
C 706
C 707
C 708
C 709
C 710
C 711
C 712
C 713
C 714
C 715
C 716
C 717
C 718
C 719
C 720
C 721
C 722
C 723
C 724
C 725
C 726
C 727
C 728
C 729
C 730
C 731
C 732
C 733
C 734
C 735
C 736
C 737
C 738
C 739
C 740
C 741
C 742
C 743
C 744
C 745
C 746
C 747
C 748
C 749
C 750
C 751
C 752
C 753
C 754
C 755
C 756
C 757
C 758
C 759
C 760
C 761
C 762
C 763
C 764
C 765
C 766
C 767
C 768
C 769
C 770
C 771
C 772
C 773
C 774
C 775
C 776
C 777
C 778
C 779
C 780
C 781
C 782
C 783
C 784
C 785
C 786
C 787
C 788
C 789
C 790
C 791
C 792
C 793
C 794
C 795
C 796
C 797
C 798
C 799
C 800
C 801
C 802
C 803
C 804
C 805
C 806
C 807
C 808
C 809
C 810
C 811
C 812
C 813
C 814
C 815
C 816
C 817
C 818
C 819
C 820
C 821
C 822
C 823
C 824
C 825
C 826
C 827
C 828
C 829
C 830
C 831
C 832
C 833
C 834
C 835
C 836
C 837
C 838
C 839
C 840
C 841
C 842
C 843
C 844
C 845
C 846
C 847
C 848
C 849
C 850
C 851
C 852
C 853
C 854
C 855
C 856
C 857
C 858
C 859
C 860
C 861
C 862
C 863
C 864
C 865
C 866
C 867
C 868
C 869
C 870
C 871
C 872
C 873
C 874
C 875
C 876
C 877
C 878
C 879
C 880
C 881
C 882
C 883
C 884
C 885
C 886
C 887
C 888
C 889
C 890
C 891
C 892
C 893
C 894
C 895
C 896
C 897
C 898
C 899
C 900
C 901
C 902
C 903
C 904
C 905
C 906
C 907
C 908
C 909
C 910
C 911
C 912
C 913
C 914
C 915
C 916
C 917
C 918
C 919
C 920
C 921
C 922
C 923
C 924
C 925
C 926
C 927
C 928
C 929
C 930
C 931
C 932
C 933
C 934
C 935
C 936
C 937
C 938
C 939
C 940
C 941
C 942
C 943
C 944
C 945
C 946
C 947
C 948
C 949
C 950
C 951
C 952
C 953
C 954
C 955
C 956
C 957
C 958
C 959
C 960
C 961
C 962
C 963
C 964
C 965
C 966
C 967
C 968
C 969
C 970
C 971
C 972
C 973
C 974
C 975
C 976
C 977
C 978
C 979
C 980
C 981
C 982
C 983
C 984
C 985
C 986
C 987
C 988
C 989
C 990
C 991
C 992
C 993
C 994
C 995
C 996
C 997
C 998
C 999
C 1000

C 400 J = 1, NH
      IF (J.GI. NH) GO TO 400
C 401
C 402
C 403
C 404
C 405
C 406
C 407
C 408
C 409
C 410
C 411
C 412
C 413
C 414
C 415
C 416
C 417
C 418
C 419
C 420
C 421
C 42
```

```

C 11 PM(1,1,2) = 0
C 12 RM(1,1,3) = 0
C 13 RM(1,1,4) = 0
C 14 C
C 15 SHORT PROMPTS FOR USER INITIATION RATES AT FIRST USER
C 16 C
C 17 IF (NUM .EQ. 1) GO TO 515
C 18 C
C 19 TO 510 J = 2, NUM
C 20 C
C 21 WRITE (6,505) NMAM(J), NMAM(1)
C 22 FORMAT (' ', A8, ' AT ', A6)
C 23 CALL FPEAT(5, 'R:', RM(1,J,1))
C 24 RM(1,J,2) = 0
C 25 RM(1,J,3) = 0
C 26 RM(1,J,4) = 0
C 27 510 CONTINUE
C 28 C
C 29 INITIATION RATES AT OTHER MODES
C 30 C
C 31 515 CONTINUE
C 32 IF = NCD + NRCF
C 33 IF (11 .EQ. 1) GO TO 510
C 34 C
C 35 DO 525 I = 2, 11
C 36 C
C 37 TO 520 J = 1, NUM
C 38 WRITE (6,505) NMAM(J), NMAM(1)
C 39 CALL FPEAT(5, 'R:', RM(1,J,1))
C 40 RM(1,J,2) = 0
C 41 RM(1,J,3) = 0
C 42 RM(1,J,4) = 0
C 43 520 CONTINUE
C 44 C
C 45 525 CONTINUE
C 46 C
C 47 INITIATION RATES & REVIEW PROBABILITIES FOR MESSAGES
C 48 FROM PATTALIONS TO BRIGADES
C 49 C
C 50 CONTINUE
C 51 IF (11 .EQ. MP) GO TO 560
C 52 II = 11 + 1
C 53 WRITE (6,505) NMAM(1), NMAM(11)
C 54 CALL FPEAT(5, 'R:', RM(11,1,1))
C 55 RM(11,1,2) = 0
C 56 C
C 57 FIRST LONG PROMPT FOR REVIEW PROBABILITIES
C 58 C
C 59 WRITE (6,470) NMAM(1), NMAM(11)
C 60 CALL FPEAT(5, 'R:', RM(11,1,1))
C 61 IF (RM(11,1,3) .GT. 1.0 .OR. RM(11,1,3) .LT. 0.0) CALL CHECK3(RM(
C 62 11,1,3))
C 63 WRITE (6,475) NMAM(1), NMAM(11)
C 64 CALL FPEAT(5, 'R:', RM(11,1,4))
C 65 IF (RM(11,1,4) .GT. 1.0 .OR. RM(11,1,4) .LT. 0.0) CALL CHECK3(RM(
C 66 11,1,4))
C 67 C
C 68 SHORT PROMPTS FOR FIRST PATTALION
C 69 C
C 70 C

```

<PAGE 11>

>>>> MAIN PROGRAM <<<<

/// FILE:CREATE ///

<PAGE 11>

```

-----
| DATE:02-05-80,14:40 | OWNER:IOSH | FILE:CREATF |
-----

```

400	IF (NUM -FO. 1) GO TO 545	400
401	DO 540 J = 2, NUM	401
402	WRITE (6,535) NMAN(J), NMAN(11)	402
403	FORMAT (' RATE, PROB OF DELETE, ALTER UP MSG ', AB, ' AT ',	403
404	AP)	404
405	CALL FREAL(5, 'JR:', RM(11,J,1), RM(11,J,3), RM(11,J,4))	405
406	IF (RM(11,J,3) -GT. 1.0 -OR. RM(11,J,3) -LT. 0.0)	406
407	CALL CHECK3(RM(11,J,1))	407
408	IF (RM(11,J,4) -GT. 1.0 -OR. RM(11,J,4) -LT. 0.0)	408
409	CALL CHECK3(RM(11,J,4))	409
410	RM(11,J,2) = 0	410
411	540 CONTINUE	411
412	C	412
413	SHORT PROMPTS FOR OTHER BATTALIONS	413
414	545 CONTINUE	414
415	IF (11 -FO. NR) GO TO 560	415
416	II = II + 1	416
417	DO 555 I = II, NH	417
418	TO 550 J = 1, NUM	418
419	WRITE (6,535) NMAN(J), NMAN(11)	419
420	CALL FREAL(5, 'JR:', RM(1,J,1), RM(1,J,3), RM(1,J,4))	420
421	IF (RM(1,J,3) -GT. 1.0 -OR. RM(1,J,3) -LT. 0.0)	421
422	CALL CHECK3(RM(1,J,1))	422
423	IF (RM(1,J,4) -GT. 1.0 -OR. RM(1,J,4) -LT. 0.0)	423
424	CALL CHECK3(RM(1,J,4))	424
425	RM(1,J,2) = 0	425
426	550 CONTINUE	426
427	C	427
428	555 CONTINUE	428
429	C	429
430	TOS GENERATED MESSAGES	430
431	FIRST LONG PROMPT FOR 1ST TOS MESSAGE PROPORTIONALITY	431
432	CONSTANTS, 1ST USER	432
433	IF (NUM -FO. NM) GO TO 585	433
434	JJ = NM + 1	434
435	WRITE (6,545) NMAN(JJ), NMAN(11)	435
436	CALL FREAL(5, '2R:', RM(1,JJ,1), RM(1,JJ,2))	436
437	RM(1,JJ,3) = 0	437
438	RM(1,JJ,4) = 0	438
439	C	439
440	SHORT PROMPTS FOR PROPORTIONALITY CONSTANTS, 1ST USER	440
441	IF (JJ -FO. NM) GO TO 570	441
442	II = NM + 2	442
443	DO 565 J = 1, NM	443
444	WRITE (6,505) NMAN(J), NMAN(11)	444
445	CALL FREAL(5, '2R:', RM(1,J,1), RM(1,J,2))	445
446	RM(1,J,3) = 0	446
447	RM(1,J,4) = 0	447
448	C	448
449	560 CONTINUE	449
450	C	450
451	565 CONTINUE	451
452	C	452
453	570 CONTINUE	453
454	C	454
455	575 CONTINUE	455
456	C	456
457	580 CONTINUE	457
458	C	458
459	585 CONTINUE	459
460	C	460
461	590 CONTINUE	461
462	C	462
463	595 CONTINUE	463
464	C	464
465	600 CONTINUE	465
466	C	466
467	605 CONTINUE	467
468	C	468
469	610 CONTINUE	469
470	C	470
471	615 CONTINUE	471
472	C	472
473	620 CONTINUE	473
474	C	474
475	625 CONTINUE	475
476	C	476
477	630 CONTINUE	477
478	C	478
479	635 CONTINUE	479
480	C	480
481	640 CONTINUE	481
482	C	482
483	645 CONTINUE	483
484	C	484
485	650 CONTINUE	485
486	C	486
487	655 CONTINUE	487
488	C	488
489	660 CONTINUE	489
490	C	490
491	665 CONTINUE	491
492	C	492
493	670 CONTINUE	493
494	C	494
495	675 CONTINUE	495
496	C	496
497	680 CONTINUE	497
498	C	498
499	685 CONTINUE	499
500	C	500
501	690 CONTINUE	501
502	C	502
503	695 CONTINUE	503
504	C	504
505	700 CONTINUE	505
506	C	506
507	705 CONTINUE	507
508	C	508
509	710 CONTINUE	509
510	C	510

//// PLF:CRATE ////

>>>> MAIN PROGRAM <<<<

&lt;PAGE 11&gt;

```

697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754

565 CONTINUE
C
C      SHORT INCRPTS FOR OTHER NODES
C
570 CONTINUE
IF (NF.FO. 1) GO TO 595
C
DO 580 I = 2, NH
C
    IO 575 J = JJ, NH
    WRITE (6,505) NHAM(J), RHAM(I)
    CALL FREQ(5, '2R:', GPD(1,1), RH(I,J,2))
    RH(I,J,3) = 0
    RH(I,J,4) = 0
575 CONTINUE
C
580 CONTINUE
C
C      ***** ENGINEERING DATA *****
C
C      TASK ACCESS TIMES
C
585 CONTINUE
WRITE (6,590)
590 FORMAT (' ENTER MEAN DATA DISK ACCESS TIME IN MILLISEC(S)')
CALL FREQ(5, 'R:', DD(2))
WRITE (6,595)
595 FORMAT (' ENTER MEAN MESSAGE DISK ACCESS TIME IN MILLISEC(S)')
CALL FREQ(5, 'R:', DD(1))
IF (TCS.NP. 1) GO TO 625
C
C      TCS DATA
C
WRITE (6,600)
600 FORMAT (' ENTER CPU TIME IN MILLISEC(S) AT A TCS TO ORIGINATE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(1,1), GPD(1,2))
IF (IV.FO. 1) GO TO 605
WRITE (6,605)
605 FORMAT (' ENTER CPU TIME IN MILLISEC(S) AT A TCS TO SEND, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(2,1), GPD(2,2))
610 FORMAT (' ENTER CPU TIME IN MILLISEC(S) AT A TCS TO RECEIVE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(3,1), GPD(3,2))
WRITE (6,615)
615 FORMAT (' ENTER CPU TIME IN MILLISEC(S) AT A TCS TO TEMPLATE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(4,1), GPD(4,2))
WRITE (6,620)
620 FORMAT (' ENTER CPU TIME IN MILLISEC(S) AT A TCS TO TERMINATE, ',
1 'PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED)')
CALL FREQ(5, '2R:', GPD(5,1), GPD(5,2))
GO TO 635
625 CONTINUE
C
DO 630 I = 1, 5

```

&lt;PAGE 11&gt;

///// FILE:CREATE /////

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

&lt;PAGE 11&gt;

```

<PAGE 14>
1 DATE:02-05-80, 14:40 CREF:IDSH FILE:CREATE
1
755 GPD(1,1) = 0.
756 GPD(1,2) = 0.
757
758 610 CONTINUE
759
760 635 CONTINUE
761
762 TCT DATA
763
764
765 IF (TCT.NF. 2) GO TO 660
766 WRITE (6,600)
767 640 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO ORIGINATE, ',
768 1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
769 CALL FREAD(5, '2R:', GPD(7,1), GPD(7,2))
770 IF (IV.FO. 1) GO TO 715
771 WRITE (6,605)
772 645 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO SEND, ',
773 1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
774 CALL FREAD(5, '2R:', GPD(8,1), GPD(8,2))
775 WRITE (6,650)
776 650 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO RECEIVE, ',
777 1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
778 CALL FREAD(5, '2R:', GPD(9,1), GPD(9,2))
779 WRITE (6,655)
780 655 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT A TCT TO TERMINATE, ',
781 1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
782 CALL FREAD(5, '2R:', GPD(10,1), GPD(10,2))
783 GO TO 670
784 660 CONTINUE
785
786 DO 665 I = 1, 4
787 GPD(I + 6,1) = 0.
788 GPD(I + 6,2) = 0.
789 665 CONTINUE
790
791 IF (IV.FO. 1) GO TO 720
792 670 CONTINUE
793
794 TFP DATA
795
796 675 WRITE (6,675)
797 675 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT THE PER TO INITIATE, ',
798 1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
799 CALL FREAD(5, '2R:', GPD(11,1), GPD(11,2))
800 WRITE (6,680)
801 680 FORMAT (' ENTER CPU TIME IN MILLISECONDS AT THE PER TO TERMINATE, ',
802 1 PER MSG AND PER CHARACTER, (2 NUMBERS NEEDED) ')
803 CALL FREAD(5, '2R:', GPD(6,1), GPD(6,2))
804 GO TO 730
805
806 SHORT PROMPTS FOR TCS DATA
807
808 685 CONTINUE
809 WRITE (6,690)
810 690 FORMAT (' TO SEND')
811 CALL FREAD(5, '2R:', GPD(2,1), GPD(2,2))
812 WRITE (6,695)
813 695 FORMAT (' TO RECEIVE')
814 CALL FREAD(5, '2R:', GPD(3,1), GPD(3,2))
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

&lt;PAGE 14&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:CREATE /////

&lt;PAGE 14&gt;



&lt;PAGE 16&gt;

DATE:02-05-80,14:40 CMMER:IOSU FILE:CREATE

&lt;PAGE 16&gt;

15N

```

871 C MESSAGE REQUIREMENTS
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C

770 CONTINUE
DO 790 I = 1, NM
  WRITE (6,775) NMAM(I)
  FORMAT (' ENTER DEP PROCESSING IN MILLISECONDS TIME FOR MSG ',
    1)
  CALL FREAT(5, 'R:', MD(I,2))
  WRITE (6,780) NMAM(I)
  FORMAT (' ENTER MSG DISK READS/Writes FOR MSG ', A9)
  CALL FREAT(5, 'R:', MD(I,3))
  WRITE (6,785) NMAM(I)
  FORMAT (' ENTER DATA DISK READS/Writes FOR MSG ', A9)
  CALL FREAT(5, 'R:', MD(I,4))
  IF (IIV.EC. 1 .AND. NM .GT. 1) GO TO 795
790 CONTINUE
GO TO 910
C
C SHORT PROMPTS FOR MESSAGE REQUIREMENTS
C
795 CONTINUE
DO 805 I = 2, NM
  WRITE (6,800) NMAM(I)
  FORMAT (' DEP TIME, MSG DISK R/W, DATA DISK R/W, FOR MSG ',
    1)
  CALL FREAT(5, 'R:', MD(I,2), MD(I,3), MD(I,4))
805 CONTINUE
C
C NUMBER OF TEMPLATES
C
810 CONTINUE
IF (NCT.EC. 0) GO TO 820
DO 815 I = 1, NCT
  ED(I) = 0
815 CONTINUE
C
820 CONTINUE
IF (NPOF.EC. 0) GO TO 835
JJ = NCT + 1
JK = NCT + NDDF
C
830 CONTINUE
DO 835 I = JJ, JK
  WRITE (6,825) NMAM(I)
  FORMAT (' ENTER NUMBER OF TEMPLATES AT ', A9)
  CALL FREAT(5, 'I:', ED(I))
  IF (IIV.EC. 1 .AND. NDDF .GT. 1) GO TO 845
835 CONTINUE
C
840 CONTINUE
WRITE (6,840)
840 FORMAT (' ENTER NUMBER OF TEMPLATES AT THE FPP')
CALL FREAT(5, 'I:', ED(NR + 1))
GO TO 870
C

```

&lt;PAGE 16&gt;

///// FILE:CREATE /////

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

&lt;PAGE 16&gt;

```

<PAGE 17>
DATE:02-05-80,14:40 CUNER:TDSH IIL:CMFATE
ISN
SHORT PROMPTS FOR # OF TEMPLATES
895 CONTINUE
IP (MPCF -FO. 1) GO TO 860
JJ = JJ + 1
DO 955 I = JJ, JK
  WRITE (6,950) JHAN(I)
  FORMAT (' AT ', A5)
  CALL FREQ(5, '1:', PL(NH + 1))
  CALL FREQ(5, '1:', PL(I))
955 CONTINUE
960 CONTINUE
  WRITE (6,865)
  FORMAT (' AT FFF ')
  CALL FREQ(5, '1:', PL(NH + 1))
970 CONTINUE
  IP (MCH + NDDF -EO. NH) GO TO 890
  JK = JK + 1
DO 975 I = JK, NR
  PR(I) = 0
975 CONTINUE
***** ENVIRONMENTAL DATA *****
      VOICE COMPETITION
980 CONTINUE
DO 895 I = 1, NC
  WRITE (6,895) CMAR(I)
  FORMAT (' ENTER PROPORTION OF TIME CHANNEL ', AR, ' IS ',
    ' PING USED FOR VOICE TRANSMISSIONS')
  CALL FREQ(5, 'R:', CL(I, 1))
  IF (CM(I, 1) -GT. 1.0 .OR. CM(I, 3) -LT. 0.0) CALL CHECK4(CM(I,
    1))
  IP (LV -FO. 1 .AND. NC -GT. 1) GO TO 895
990 CONTINUE
GO TO 910
      SHORT PROMPTS FOR VOICE COMPETITION INFO
995 CONTINUE
DO 905 I = 2, NC
  WRITE (6,900) CMAR(I)
  FORMAT (' CHANNEL ', AR)
  CALL FREQ(5, 'R:', CM(I, 1))
  IF (CM(I, 1) -GT. 1.0 .OR. CM(I, 3) -LT. 0.0) CALL CHECK4(CM(I,
    1))
905 CONTINUE
      ERROR RATES: TO 6 FROM FOP
910 CONTINUE
  JN = MCH + NDDF

```

&lt;PAGE 17&gt;

///// PEEP:CREATE /////

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

&lt;PAGE 17&gt;

```

C
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
```

```

1045 2      LR, GPD, IICS, IICU)
1046 C
1047 C      ***** WRITE OUTPUT FILE HERE *****
1048 C
1049 C      WRITE (6,900)
1050 900 FORMAT (' IF YOU WANT TO CREATE THE OUTPUT FILE ENTER, * A *100')
1051 CALL FREAD(5, '1:', IWF)
1052 IF (IWF .NE. 0) GO TO 1075
1053 C
1054 C      INDEXING INFORMATION AND COUNTS
1055 C
1056 C      WRITE (2,905) NR, NP, NW, NW, NC, NP, NI, NOW, NDOZ, IFG
1057 905 FORMAT (10(I1,1X))
1058 C
1059 C      ROUTE INFORMATION
1060 C
1061 C      DO 995 I = 1, NR
1062 995 WRITE (2,900) RMAN(I), RE(I,1), RP(I,2), IPD(I)
1063 900 FORMAT (A9, 3(1X,1))
1064 995 CONTINUE
1065 C
1066 C      MESSAGE INFORMATION
1067 C
1068 C      DO 1005 I = 1, NW
1069 1005 WRITE (2,1000) MNAME(I), MP(I,1), MP(I,2), MD(I,1), MD(I,2),
1070 1 MD(I,3), MD(I,4)
1071 1000 FORMAT (A4, 2(1X,1), 4(1X,F9.3))
1072 1005 CONTINUE
1073 C
1074 C      CHANNEL INFORMATION
1075 C
1076 C      DO 1015 I = 1, NC
1077 1015 WRITE (2,1010) CNAME(I), CD(I,1), CD(I,2), CD(I,3), CD(I,4)
1078 1010 FORMAT (A6, 4(1X,F9.3))
1079 1015 CONTINUE
1080 C
1081 C      NUMBER OF ITERATIONS
1082 C
1083 C      DO 1025 I = 1, NP
1084 1025 WRITE (2,1020) ID(I)
1085 1020 FORMAT (2X, 1)
1086 1025 CONTINUE
1087 C
1088 C      LINK DATA
1089 C
1090 C      DO 1035 I = 1, NI
1091 1035 WRITE (2,1030) IP(I,1), IP(I,2), LP(I,1), LP(I,2), LD(I,1), LD(I,2)
1092 1030 FORMAT (3(1,1X), 2(E12.6,1X))
1093 1035 CONTINUE
1094 C
1095 C      ROUTE MESSAGE ARRAY
1096 C
1097 C      IF (IFG .EQ. 1) WRITE (2,1040)
1098 1040 FORMAT (' ', 50(' '))
1099 IF (IFG .EQ. 1) GO TO 1060
1100 C
1101 C      DO 1055 I = 1, NW
1102 1055
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112

```

&lt;PAGE 17&gt;

&lt;PAGE 19&gt;

DATE:02-05-80,10:40 CNAME:IDSH FILE:CREATE

ISM

&gt;&gt;&gt;&gt; DATA PROGRAM &lt;&lt;&lt;&lt;

///// FILE:CREATE /////

&lt;PAGE 19&gt;

&lt;PAGE 19&gt;

```

1103      DO 1050 J = 1, NH
1104          WRITE (2,1045) RM(I,J,1), P*(I,J,2), RM(I,J,3),
1105              RM(I,J,4)
1106          FORMAT (4(F9.3,1X))
1107      CONTINUE
1108  C
1109  C
1110  C      RISK ACCESS TIMES
1111  C
1112  C
1113  C
1114  C      DO 1060 CONTINUE
1115  C          WRITE (2,1065) DD(1), DD(2)
1116  C          FORMAT (2(F9.3,1X))
1117  C
1118  C      GENERAL IPROCESSOR DATA
1119  C
1120  C          DO 1070 I = 1, 11
1121  C              WRITE (2,1065) GPD(I,1), GPD(I,2)
1122  C          CONTINUE
1123  C
1124  C      1075 CONTINUE
1125  C          END

```

&lt;PAGE 20&gt;

```

1  DATE: 02-05-90, 14:40  CYNSE: 1DSH  FILE: CREATP

```

ISM

702

703

704

705

706

707

708

709

710

711

712

713

714

&lt;PAGE 20&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE: CREATP /////

&lt;PAGE 20&gt;

&lt;PAGE 21&gt;

1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144

&lt;PAGE 21&gt;

1 DATE:02-05-00, 14:40 CMMR:TOSH FILE:CRPATF 1  
ISM

\*\*\*\*\* SUBROUTINES \*\*\*\*\*

THIS SUBROUTINE CHECKS THE NAME OF THE PROCESSOR TYPE,  
THE CORRECT NAME IS RETURNED AS VARIABLE "TYPE"

SUBROUTINE CHECK(TYPE)

10 CONTINUE  
INTEGER TYPE, TCT /'TCT ', TCS /'TCS '  
IF (TYPE .EQ. TCT) RETURN  
IF (TYPE .EQ. TCS) RETURN  
WRITE (6,15) TYPE  
15 FORMAT (' THERE IS NO PROCESSOR NAMED ', A0, 'TRY AGAIN')  
CALL FPEAD(5, 'S:', TYPE, 4)  
GO TO 10  
END

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

&lt;PAGE 21&gt;

///// FILE:CREATE /////

>>>> SUBROUTINE CRPCN <<<<

&lt;PAGE 21&gt;

<PAGE 22>  
15N

DATE: 07-05-80, 14:40 GMMER: 1958 FILE: CREATE

```

1185 C
1186 C
1187 C
1188 C
1189 C
1190 C
1191 C
1192 C
1193 C
1194 C
1195 C
1196 C
1197 C
1198 C
1199 C
1200 C
1201 C
1202 C
1203 C
1204 C
1205 C
1206 C
1207 C
1208 C
1209 C
1210 C
1211 C
1212 C
1213 C
1214 C
1215 C
1216 C
1217 C
1218 C
1219 C
1220 C
1221 C
1222 C
1223 C
1224 C
1225 C
1226 C
1227 C
1228 C
1229 C
1230 C
1231 C
1232 C
1233 C
1234 C
1235 C
1236 C
1237 C
1238 C
1239 C
1240 C
1241 C
1242 C
1243 C
1244 C
1245 C
1246 C
1247 C
1248 C
1249 C
1250 C
1251 C
1252 C
1253 C
1254 C
1255 C
1256 C
1257 C
1258 C
1259 C
1260 C
1261 C
1262 C
1263 C
1264 C
1265 C
1266 C
1267 C

```

THIS SUBROUTINE CHECKS THE NAME OF THE CHANNEL "NIN" AND  
RETURNS THE NUMBER OF THAT CHANNEL IN VARIABLE "NIN1"

SUBROUTINE CHECK1(NIN, CNAM, NC, NIN1)

REAL\*8 NIN, CNAM(10)

10 CONTINUE

DO 15 I = 1, NC  
IF (NIN .EQ. CNAM(I)) GO TO 15  
NIN1 = I  
RETURN

15 CONTINUE

WRITE (6,20) NIN  
20 PRINT (' THERE IS NO CHANNEL NAMED ', AD, ' TRY AGAIN')  
CALL FREAD('S', 'S:', NIN, B)  
GO TO 10  
END

&lt;PAGE 22&gt;

&gt;&gt;&gt;&gt; SUBROUTINE CHECK1 &lt;&lt;&lt;&lt;

///// FILE:CREATE /////

&lt;PAGE 22&gt;

&lt;PAGE 23&gt;

15N

DATE:07-05-00,10:40 CUMER:105H FILE:CREATF

&lt;PAGE 24&gt;

1164  
1165  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195

THIS SUBROUTINE CHECKS THE NAME OF THE MESSAGE "HOUT" AND  
RETURNS THE NUMBER OF THAT MESSAGE IN VARIABLE "MOUT1"

SUBROUTINE CHECK2(MCUT, NM, MNAM, MOUT1)

REAL\*8 MOUT, MNAM(50), ZERO  
DATA ZERO /0.0/  
10 CONTINUE  
IF (MCUT.EQ.ZERO) GO TO 25

DO 15 I = 1, NM  
IF (MNAM(I).NE.HOUT) GO TO 15  
MOUT1 = I  
RETURN  
15 CONTINUE

WRITE (6,20) MCUT  
20 FORMAT (' THERE IS NO MSG NAMED ', A8, 'TRY AGAIN')  
CALL FREAD(5, 'S:', HOUT, 0)  
GO TO 10  
25 CONTINUE  
MOUT1 = 0  
RETURN  
END

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18

&lt;PAGE 23&gt;

&gt;&gt;&gt;&gt; SUBROUTINE CHECK2 &lt;&lt;&lt;&lt;

///// FILE:CREATF /////

&lt;PAGE 24&gt;

```

1196 C
1197 C
1198 C
1199 C
1200 C
1201 C
1202 C
1203 C
1204 C
1205 C
1206 C
1207 C
1208 C
1209 C
1210 C
1211 C
1212 C

      THIS SUBROUTINE REPLACES PROBABILITIES WHICH HAVE
      BEEN ENTERED OUT OF THE RANGE 0-1 AND RETURNS THE
      EXCESS VALUE IN VARIABLE "PROB".

      SUBROUTINE CHECK1(PRCB)
      10 CONTINUE
      15 WRITE (6,15)
      15 FORMAT (' THE PROBABILITY IS OUT OF THE RANGE 0-1, TRY AGAIN')
      CALL PFAD(5, 'R:', PRCH)
      IF (PROB .LT. 1.0 .AND. PRCB .GT. 0.0) RETURN
      GO TO 10
      END

```

1  
2  
3  
4  
5  
6  
7  
8

<PAGE 24>

>>>> SUBROUTINE CHECK1 <<<<

///// FILE:CREATE /////

&lt;PAGE 25&gt;

1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229

1 DATE:02-05-80, 14:40 CTYPE:DSH FILE:CREATF  
ISM

&lt;PAGE 25&gt;

THIS SUBROUTINE REPLACES PROPORTIONS WHICH HAVE  
BEEN ENTERED OUT OF THE RANGE 0-1 AND RETURNS THE  
PROPER VALUE IN VARIABLE "PROP".

SUBROUTINE CHECK4(PRCF)

10 CONTINUE

WRITE (6,15)

15 FORMAT (' THE PROPORTION IS OUT OF THE RANGE 0-1, TRY AGAIN')

CALL READ(5, 'R:', PRCP)

IF (PRCP .LT. 1.0 .AND. PRCP .GT. 0.0) RETURN

GO TO 10

END

1  
2  
3  
4  
5  
6  
7  
8

&lt;PAGE 25&gt;

//// FILE:CREATF ////

>>>> SUBROUTINE CHECK4 <<<<

&lt;PAGE 25&gt;

&lt;PAGE 26&gt;

&lt;PAGE 26&gt;

LSM

DATE:02-05-00,14:40 CUSER:IDSH FILE:CHDATE

```

1210 C
1211 C
1212 C
1213 C
1214 C
1215 C
1216 C
1217 C
1218 C
1219 C
1220 C
1221 C
1222 C
1223 C
1224 C
1225 C
1226 C
1227 C
1228 C
1229 C
1230 C
1231 C
1232 C
1233 C
1234 C
1235 C
1236 C
1237 C
1238 C
1239 C
1240 C
1241 C
1242 C
1243 C
1244 C
1245 C
1246 C
1247 C
1248 C
1249 C
1250 C
1251 C
1252 C
1253 C
1254 C
1255 C
1256 C
1257 C
1258 C
1259 C
1260 C
1261 C
1262 C
1263 C
1264 C
1265 C
1266 C
1267 C
1268 C
1269 C
1270 C
1271 C
1272 C
1273 C
1274 C
1275 C
1276 C
1277 C
1278 C
1279 C
1280 C
1281 C
1282 C
1283 C
1284 C
1285 C
1286 C
1287 C

      THIS SUBROUTINE DISPLAYS THE OUTPUT FILE IN AN
      EASY TO READ FORMAT. THIS SUBROUTINE IS ALSO
      FORMED AS PART OF PROGRAM DISPLAY.

      SUBROUTINE DISP(NR, NM, NN, NNM, NC, NP, NL, NMM, NNDP, IFG,
1      RNAM, RP, LCP, RMAP, RD, PD, CNAME, CD, PD, LP, LD, RM,
2      DD, GPD, ITCS, LTCI)

      REAL*8 RNAM(50), RNM(50), CNAME(50)
      REAL LP(50,2), RD(50,4)
      INTEGER NP(50,2), PD(50), LP(50,3)
      DIMENSION DD(2), GPD(11,2), CD(10,4), MP(50,2), RM(50,50,4),
1      IPD(50)

      WRITE SYSTEM CONFIGURATION HEADERS

      WRITE (6,10)
10  FORMAT ('- ', 37X, 50(' '), /, 30X, '-----', 40X, '-----')
15  FORMAT (30X, '-----', 40X, '-----', /, 30X, 50(' '), /)
      WRITE (6,20)
20  FORMAT (30X, '-----', 10X, 'SYSTEM CONFIGURATION', 10X, '-----')
      WRITE (6,15)
      IF (NRCE - PD - 0) GO TO 75

      CONFIGURATION OF BRIGADES

      IT = NCB + 1
      JJ = RCEC + NCB
      WRITE (6,25)
25  FORMAT ('0', 50X, '--- NAME/VER BRIGADES ---', //, 22X, 'UNIT',
1      '16X, 'COMMUNICATIONS', 11X, 'PROCESSOR', 16X, 'NUMBER',
2      '0', /, 22X, 'NAME', 17X, 'LINK W/ PEP', 17X, 'TYPE',
3      '19X, 'RELATES', /)

      DO 45 I = II, JJ
      IF (IPD(I) - EQ. 1) GO TO 35
      WRITE (6,30) RNAM(I), CNAME(LP(I,2)), PD(I)
      FORMAT (22X, A8, 15X, A8, 19X, 'TCT', 22X, 12)
      GO TO 45
35  WRITE (6,40) RNAM(I), CNAME(LP(I,2)), PD(I)
40  FORMAT (22X, A8, 15X, A8, 19X, 'TCS', 22X, 12)
      GO TO 45
45  CONTINUE

      CONFIGURATION OF BATTALIONS

      IF (JJ - PD - NR) GO TO 75
      WRITE (6,50)
50  FORMAT ('0', 40X, '--- NAME/VER BATTALIONS ---', //, 22X, 'UNIT',
1      '16X, 'COMMUNICATIONS', 15X, 'PARENT', 17X, 'PROCESSOR', /,
2      '22X, 'NAME', 16X, 'LINK W/ BRIGADE', 14X, 'BRIGADE', 18X,
3      'TYPE', /)
      JJ = JJ + 1
      ISPACE = IP(JJ,3)
      GO TO 75
75  CONTINUE

```

&lt;PAGE 26&gt;

///// FILE:CHDATE /////

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 26&gt;

&lt;PAGE 27&gt;

-----  
 1 DATE:02-05-80,14:40 USER:IOSH FILE:CREATE  
 -----

&lt;PAGE 27&gt;

158

```

1298 DO 70 I = 1, NR
1299   IF (LP(I,1) .NE. 1) WRITE (6,115)
1300   ISPACE = IP(I,3)
1301   IF (IP(I,1) .EQ. 1) GO TO 60
1302   WRITE (6,115) READ(I), CHAM(LP(I,2)), RNAME(LP(I,3))
1303   FORMAT (22X, A6, 16X, A9, 18X, A9, 11X, 'TCT')
1304   GO TO 70
1305 60 WRITE (6,115) RNAME(I), CHAM(LP(I,2)), RNAME(LP(I,3))
1306   FORMAT (22X, A6, 16X, A9, 18X, A9, 17X, 'TCS')
1307 70 CONTINUE
1308 C
1309 75 CONTINUE
1310 C
1311 C
1312 C
1313 C
1314 C
1315 C
1316 C
1317 C
1318 C
1319 C
1320 C
1321 C
1322 C
1323 C
1324 C
1325 C
1326 C
1327 C
1328 C
1329 C
1330 C
1331 C
1332 C
1333 C
1334 C
1335 C
1336 C
1337 C
1338 C
1339 C
1340 C
1341 C
1342 C
1343 C
1344 C
1345 C
1346 C
1347 C
1348 C
1349 C
1350 C
1351 C
1352 C
1353 C
1354 C
1355 C
1356 C
1357 C
1358 C
1359 C
1360 C
1361 C
1362 C
1363 C
1364 C
1365 C
1366 C
1367 C
1368 C
1369 C
1370 C
1371 C
1372 C
1373 C
1374 C
1375 C
1376 C
1377 C
1378 C
1379 C
1380 C
1381 C
1382 C
1383 C
1384 C
1385 C
1386 C
1387 C
1388 C
1389 C
1390 C
1391 C
1392 C
1393 C
1394 C
1395 C
1396 C
1397 C
1398 C
1399 C
1400 C
1401 C
1402 C
1403 C
1404 C
1405 C
1406 C
1407 C
1408 C
1409 C
1410 C
1411 C
1412 C
1413 C
1414 C
1415 C
1416 C
1417 C
1418 C
1419 C
1420 C
1421 C
1422 C
1423 C
1424 C
1425 C
1426 C
1427 C
1428 C
1429 C
1430 C
1431 C
1432 C
1433 C
1434 C
1435 C
1436 C
1437 C
1438 C
1439 C
1440 C
1441 C
1442 C
1443 C
1444 C
1445 C
1446 C
1447 C
1448 C
1449 C
1450 C
1451 C
1452 C
1453 C
1454 C
1455 C
1456 C
1457 C
1458 C
1459 C
1460 C
1461 C
1462 C
1463 C
1464 C
1465 C
1466 C
1467 C
1468 C
1469 C
1470 C
1471 C
1472 C
1473 C
1474 C
1475 C
1476 C
1477 C
1478 C
1479 C
1480 C
1481 C
1482 C
1483 C
1484 C
1485 C
1486 C
1487 C
1488 C
1489 C
1490 C
1491 C
1492 C
1493 C
1494 C
1495 C
1496 C
1497 C
1498 C
1499 C
1500 C
1501 C
1502 C
1503 C
1504 C
1505 C
1506 C
1507 C
1508 C
1509 C
1510 C
1511 C
1512 C
1513 C
1514 C
1515 C
1516 C
1517 C
1518 C
1519 C
1520 C
1521 C
1522 C
1523 C
1524 C
1525 C
1526 C
1527 C
1528 C
1529 C
1530 C
1531 C
1532 C
1533 C
1534 C
1535 C
1536 C
1537 C
1538 C
1539 C
1540 C
1541 C
1542 C
1543 C
1544 C
1545 C
1546 C
1547 C
1548 C
1549 C
1550 C
1551 C
1552 C
1553 C
1554 C
1555 C
1556 C
1557 C
1558 C
1559 C
1560 C
1561 C
1562 C
1563 C
1564 C
1565 C
1566 C
1567 C
1568 C
1569 C
1570 C
1571 C
1572 C
1573 C
1574 C
1575 C
1576 C
1577 C
1578 C
1579 C
1580 C
1581 C
1582 C
1583 C
1584 C
1585 C
1586 C
1587 C
1588 C
1589 C
1590 C
1591 C
1592 C
1593 C
1594 C
1595 C
1596 C
1597 C
1598 C
1599 C
1600 C
1601 C
1602 C
1603 C
1604 C
1605 C
1606 C
1607 C
1608 C
1609 C
1610 C
1611 C
1612 C
1613 C
1614 C
1615 C
1616 C
1617 C
1618 C
1619 C
1620 C
1621 C
1622 C
1623 C
1624 C
1625 C
1626 C
1627 C
1628 C
1629 C
1630 C
1631 C
1632 C
1633 C
1634 C
1635 C
1636 C
1637 C
1638 C
1639 C
1640 C
1641 C
1642 C
1643 C
1644 C
1645 C
1646 C
1647 C
1648 C
1649 C
1650 C
1651 C
1652 C
1653 C
1654 C
1655 C
1656 C
1657 C
1658 C
1659 C
1660 C
1661 C
1662 C
1663 C
1664 C
1665 C
1666 C
1667 C
1668 C
1669 C
1670 C
1671 C
1672 C
1673 C
1674 C
1675 C
1676 C
1677 C
1678 C
1679 C
1680 C
1681 C
1682 C
1683 C
1684 C
1685 C
1686 C
1687 C
1688 C
1689 C
1690 C
1691 C
1692 C
1693 C
1694 C
1695 C
1696 C
1697 C
1698 C
1699 C
1700 C
1701 C
1702 C
1703 C
1704 C
1705 C
1706 C
1707 C
1708 C
1709 C
1710 C
1711 C
1712 C
1713 C
1714 C
1715 C
1716 C
1717 C
1718 C
1719 C
1720 C
1721 C
1722 C
1723 C
1724 C
1725 C
1726 C
1727 C
1728 C
1729 C
1730 C
1731 C
1732 C
1733 C
1734 C
1735 C
1736 C
1737 C
1738 C
1739 C
1740 C
1741 C
1742 C
1743 C
1744 C
1745 C
1746 C
1747 C
1748 C
1749 C
1750 C
1751 C
1752 C
1753 C
1754 C
1755 C
1756 C
1757 C
1758 C
1759 C
1760 C
1761 C
1762 C
1763 C
1764 C
1765 C
1766 C
1767 C
1768 C
1769 C
1770 C
1771 C
1772 C
1773 C
1774 C
1775 C
1776 C
1777 C
1778 C
1779 C
1780 C
1781 C
1782 C
1783 C
1784 C
1785 C
1786 C
1787 C
1788 C
1789 C
1790 C
1791 C
1792 C
1793 C
1794 C
1795 C
1796 C
1797 C
1798 C
1799 C
1800 C
1801 C
1802 C
1803 C
1804 C
1805 C
1806 C
1807 C
1808 C
1809 C
1810 C
1811 C
1812 C
1813 C
1814 C
1815 C
1816 C
1817 C
1818 C
1819 C
1820 C
1821 C
1822 C
1823 C
1824 C
1825 C
1826 C
1827 C
1828 C
1829 C
1830 C
1831 C
1832 C
1833 C
1834 C
1835 C
1836 C
1837 C
1838 C
1839 C
1840 C
1841 C
1842 C
1843 C
1844 C
1845 C
1846 C
1847 C
1848 C
1849 C
1850 C
1851 C
1852 C
1853 C
1854 C
1855 C
1856 C
1857 C
1858 C
1859 C
1860 C
1861 C
1862 C
1863 C
1864 C
1865 C
1866 C
1867 C
1868 C
1869 C
1870 C
1871 C
1872 C
1873 C
1874 C
1875 C
1876 C
1877 C
1878 C
1879 C
1880 C
1881 C
1882 C
1883 C
1884 C
1885 C
1886 C
1887 C
1888 C
1889 C
1890 C
1891 C
1892 C
1893 C
1894 C
1895 C
1896 C
1897 C
1898 C
1899 C
1900 C
1901 C
1902 C
1903 C
1904 C
1905 C
1906 C
1907 C
1908 C
1909 C
1910 C
1911 C
1912 C
1913 C
1914 C
1915 C
1916 C
1917 C
1918 C
1919 C
1920 C
1921 C
1922 C
1923 C
1924 C
1925 C
1926 C
1927 C
1928 C
1929 C
1930 C
1931 C
1932 C
1933 C
1934 C
1935 C
1936 C
1937 C
1938 C
1939 C
1940 C
1941 C
1942 C
1943 C
1944 C
1945 C
1946 C
1947 C
1948 C
1949 C
1950 C
1951 C
1952 C
1953 C
1954 C
1955 C
1956 C
1957 C
1958 C
1959 C
1960 C
1961 C
1962 C
1963 C
1964 C
1965 C
1966 C
1967 C
1968 C
1969 C
1970 C
1971 C
1972 C
1973 C
1974 C
1975 C
1976 C
1977 C
1978 C
1979 C
1980 C
1981 C
1982 C
1983 C
1984 C
1985 C
1986 C
1987 C
1988 C
1989 C
1990 C
1991 C
1992 C
1993 C
1994 C
1995 C
1996 C
1997 C
1998 C
1999 C
2000 C

```

&lt;PAGE 27&gt;

///// FILE:CREATE /////

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 27&gt;

&lt;PAGE 28&gt;

&lt;PAGE 28&gt;

15M

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

&lt;PAGE 28&gt;

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 28&gt;

1 DATE:02-05-80,10:40 CYNH:JDSN JILL:CRFATE

```

135 FORMAT ('0', A1, 'MESSAGE', 7X, 'ORIGINATOR', A1, 'NAME', 12X,
1 'REP', 9X, 'MSG DISK', A1, 'DATA DISK', 7X, 'MSG OUT', A1,
2 'MSG OUT', 7X, 'WAMP', 26X, 'LENGTH', 11X, 'TIME', 6X,
3 'REAR/WITES', 11, 'READS/WITES', 4X, 'SAFE ROUTE', 4X,
4 'CINR ROUTE', 16X, 'CHARACTERS', 5X, 'MILLSSEC', 5X,
5 'UPR MSG', 5X, 'UPR MSG', /)

```

USER GENERATED MESSAGES

IF (NUM -FO. 0) GO TO 165

DO 160 I = 1, NUM

IND1 = MC(I,1)

IND3 = MC(I,3)

IND4 = MC(I,4)

```

1 IF (IND1,1) -NE. 0 -AND. MP(I,2) -NE. 0 WRITE (6,140) MMAM(
1 I, IND1, MD(I,2), IND3, IND4, MMAM(MP(I,1), MMAM(MP(I,2)))
160 FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I1, 10X, A8, 7X, A8)

```

```

1 IF (MP(I,1) -NE. 0 -AND. MP(I,2) -FO. 0) WRITE (6,145) MMAM(
1 I, IND1, MD(I,2), IND3, IND4, MMAM(MP(I,1),
165 FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I3, 10X, A8, 7X, 'NONE')

```

```

1 IF (MP(I,1) -EQ. 0 -AND. MP(I,2) -NE. 0) WRITE (6,150) MMAM(
1 I, IND1, MD(I,2), IND3, IND4, MMAM(MP(I,2)))
160 FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I3, 10X, 'NONE', 11X, A8)

```

```

1 IF (MP(I,1) -EQ. 0 -AND. MP(I,2) -FO. 0) WRITE (6,155) MMAM(
1 I, IND1, MD(I,2), IND3, IND4
165 FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I3, 10X, 'NONE', 11X, 'NONE')

```

160 CONTINUE

165 CONTINUE

105 GENERATED MESSAGES

IF (NUM -FO. 0) GO TO 180

IF (NUM -NE. 0 -AND. IC -NE. 0) WRITE (6,115)

II = NUM + 1

DO 175 J = 1, NN

IND1 = MD(I,1)

IND3 = MD(I,3)

IND4 = MP(I,4)

WRITE (6,170) MMAM(I), IND1, MD(I,2), IND3, IND4

```

170 FORMAT (9X, A8, 9X, 'USER', 10X, I4, 12X, F6.3, 10X, I2, 12X,
1 I3, 10X, 'NONE', 11X, 'NONE')

```

175 CONTINUE

180 CONTINUE

WRITE PROCESSOR HEADERS

WRITE (6,10)

WRITE (6,155)

105 FORMAT ('0', '.....', 13X, 'PROCESSOR DATA', 13X, '.....')

WRITE (6,15)

&lt;PAGE 28&gt;

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 28&gt;

```

-----
| DATE: 02-05-80, 14:40 QYMER: ICSH FILE: CREATE |
-----

```

>>>> SUBROUTINE DISP <<<<<

//// FILE:CRPATE ////

&lt;PAGE 30&gt;

1 DATE:02-05-80,14:40 CMMER:IDSH FILE:CREATP  
 1

&lt;PAGE 30&gt;

158

260 FORMAT (1X, 'PRIGADE', 10X, 'MESSAGE', 9X, 'INITIATION', /, 61X,  
 1 'TYPE', 14X, 'RATE', /, 76X, '(MSG/HOUR)', /)

JJ = MOD(I, 1)  
 JK = MOD(J, 1)

DO 255 I = JJ, JK

PO 250 J = 1, NUN

IRM = RM(1,J,1)

WRITE (6,265) RMAN(I), RMAN(J), IRM

FORMAT (43X, A8, 9X, A8, 11X, 13)

CONTINUE

WRITE (6,415)

255 CONTINUE

IP (JR,PO,MR) GO TO 280

WRITE (6,415)

WRITE (6,260)

MESSAGES FROM BATTALIONS

260 FORMAT (15X, 'PATTALION', 9X, 'PARFMT', 12X, 'MESSAGE', 11X,  
 1 'INITIATION', 9X, 'FRCB OP', 12X, 'PROB OP', /, 33X,  
 2 'BRIGADE', 12X, 'TYPE', 16X, 'RATE', 12X, 'DEFLECTION', 10X,  
 3 'ATTENTION', /, 69X, '(MSG/HOUR)', 9X, 'AT REC', 11X,  
 4 'AT DOE', /)

JL = JK + 1

ISPACE = LP(JL,3)

NO 270 I = JL, MR

IP (LP(I,3) - ME, ISPACE) WRITE (6,415)

ISPACE = LP(I,3)

GO 270 J = 1, NUN

IRM = RM(1,J,1)

WRITE (6,265) RMAN(I), RMAN(LP(I,3)), RMAN(J), IRM,

RM(1,J,3), RM(1,J,4)

FORMAT (15X, A8, 10X, A8, 10X, A8, 13X, 13, 15X, F4.2,  
 15X, F4.2)

CONTINUE

WRITE (6,415)

275 CONTINUE

280 CONTINUE

IP (BCB,PO,0) GO TO 100

WRITE (6,415)

WRITE (6,265)

MESSAGES FROM CMMR USERS

285 FORMAT (43X, 'OTHER', 12X, 'MESSAGE', 9X, 'INITIATION', /, 43X,  
 1 'USPRS', 13X, 'TYPE', 14X, 'RATE', /, 76X, '(MSG/HOUR)')

NO 290 I = 1, NUN

CONTINUE

CONTINUE

CONTINUE

CONTINUE

&lt;PAGE 30&gt;

///// FILE:CREATP /////

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 30&gt;

&lt;PAGE 11&gt;

1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577

```

DO 290 J = 1, NMN
  IMM = MM(I,J,1)
  WRITE (6,245) MMAR(I), MMAR(J), IMM
290 CONTINUE
C
295 WRITE (6,415)
295 CONTINUE
C
300 CONTINUE
  IP (MMH - EQ. MP) GO TO 155
  JJ = MMH + 1
  WRITE (6,305)
C
C
C 105 GENERATED MESSAGES
C
305 FORMAT (4X, ' ** TOS GENERATED MESSAGES **', //, 43X,
1 'TERMINUS', 4X, 'MESSAGE', 7X, 'OUTPUT', 4X, 'OUTPUT', //,
2 56X, 'TYPE', 9X, 'RATIO A', 1X, 'RATIO B', /)
C
C 105 MESSAGES TO BRIGADES
C
  IP (MMDF - EQ. O) GO TO 340
  II = MMH + 1
  JJ = MCB + MMDF
C
  DO 320 I = II, JJ
    DO 315 J = JJ, MM
      WRITE (6,310) MMAR(I), MMAR(J), MM(I,J,1), MM(I,J,2)
      FORMAT (43X, A8, 4X, A8, 6X, F5.1, 5X, F5.1)
315 CONTINUE
C
320 CONTINUE
C
325 CONTINUE
C
C 105 MESSAGES TO BATTALIONS
C
  IP (MMH + MMDF - EQ. MM) GO TO 340
  WRITE (6,415)
  JJ = JJ + 1
  ISPACE = IP(JJ,3)
C
  DO 335 I = JJ, MM
    IF (IP(I,1) - NE. ISPACE) WRITE (6,415)
    ISPACE = IP(I,3)
C
    DO 330 J = JJ, MM
      WRITE (6,310) MMAR(I), MMAR(J), MM(I,J,1), MM(I,J,2)
      CONTINUE
330 CONTINUE
C
335 WRITE (6,415)
335 CONTINUE
C
340 CONTINUE
C
C 105 MESSAGES TO OTHER USERS

```

&lt;PAGE 11&gt;

////// FILE:CREATE /////

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 11&gt;

DATE:02-05-80,14:40 CMMER:105H FILE:CREATE

158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195

&lt;PAGE 32&gt;

 I DATE:02-05-80,14:40 CYMER:YOSH FILE:CREATE  
 I-----I

&lt;PAGE 32&gt;

15W

```

1578 C      IP (MCU -FO. 0) GO TC 355
1579      WRITE (6,415)
1580 C
1581 C      DO 350 I = 1, MCU
1582 C
1583 C          CO 345 J = 1J, MJ
1584 C              WRITE (6,310) RMAN(I), RMAN(J), RMAN(I,J,1), RMAN(I,J,2)
1585 C          CONTINUE
1586 C
1587 C              WRITE (6,415)
1588 C          CONTINUE
1589 C
1590 C      350 CONTINUE
1591 C
1592 C      355 CONTINUE
1593 C
1594 C          WRITE THE ERROR RATE ARRAY
1595 C
1596 C              WRITE (6,10)
1597 C              WRITE (6,360)
1598 C              FORMAT (30X, '*****', 11X, 'ERROR RATES', 15X, '*****')
1599 C              WRITE (6,15)
1600 C              WRITE (6,365)
1601 C              365 FORMAT (15X, 'SENDING', 7X, 'RECEIVING', 8X, 'ERROR',
1602 C                  1 22X, 'SENDING', 7X, 'RECEIVING', 8X, 'ERROR', /,
1603 C                  2 16X, 'MODE', 11X, 'MODE', 11X, 'PATH', 24X, 'MODE', 11X,
1604 C                  3 'MODE', 11X, 'MODE', /)
1605 C              IP (NEXP -FO. 0) GO TC 195
1606 C              JJ = MCU + MDO
1607 C              IJ = PCU + 1
1608 C
1609 C      DO 375 I = 1J, JJ
1610 C          WRITE (6,370) RMAN(I), LD(I,1), RMAN(I), LD(I,2)
1611 C          FORMAT (15X, AB, 11X, 'PPE', 11X, P5.1, 22X, 'FPE', 12X, AB,
1612 C              1 8X, P5.1)
1613 C      375 CONTINUE
1614 C
1615 C      380 CONTINUE
1616 C          IP (JA -FO. 0) GO TC 395
1617 C          WRITE (6,415)
1618 C          JJ = MCU + MDO + 1
1619 C          ISPACE = LP(JJ,3)
1620 C
1621 C      DO 390 I = 1J, MJ
1622 C          IP (LP(I,1) -NE. ISPACE) WRITE (6,415)
1623 C          ISPACE = LP(I,3)
1624 C          WRITE (6,385) RMAN(I), RMAN(LP(I,3)), LD(I,1), RMAN(LP(I,3)),
1625 C              1 RMAN(I), LD(I,2)
1626 C          385 FORMAT (15X, AB, 7X, AB, 8X, P5.1, 22X, AB, 7X, AB, 8X, P5.1)
1627 C      390 CONTINUE
1628 C
1629 C      395 CONTINUE
1630 C          IP (MCU -FO. 0) GO TC 405
1631 C          WRITE (6,415)
1632 C
1633 C      DO 400 I = 1, MCU
1634 C          WRITE (6,370) RMAN(I), LD(I,1), RMAN(I), LD(I,2)
1635 C      400 CONTINUE

```

&lt;PAGE 32&gt;

///// FILE:CREATE /////

&gt;&gt;&gt;&gt; SUBROUTINE DISP &lt;&lt;&lt;&lt;

&lt;PAGE 32&gt;

<PAGE 33>

ISM

235  
236  
237  
238  
239  
240

DATE: 07-05-80, 10:40 CMMER:IDSW FILE:CMRATF

405 CONTINUE  
WRITE (6, 410)  
410 FORMAT (' ')  
415 FORMAT (' ')  
RETURN  
END

<PAGE 31>

1636  
1637  
1638  
1639  
1640  
1641

<PAGE 33>

>>>> SUBROUTINE DISE <<<<

//// FILE:CMRATF ////

<PAGE 31>

&lt;PAGE 1&gt;

DATE:02-05-80,14:01 CYNPR:IOSH FILE:DISPLAY

ISH

```

1 C
2 C
3 C
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C

***** F - I - S - P - I - A - V *****
***** TOS FILE WRITING ROUTINE *****

THIS PROGRAM WILL READ THE TOS SIMULATION
INPUT FILE CREATED BY PROGRAM CREATE OR
PROGRAM MODIFY AND WRITE THE DATA IN AN
FAST TO REAL FORMAT.

THE INPUT FILE IS READ FROM DEVICE NUMBER 2

      DEK VRI API-3 23 JAN 80

      READ INPUT FILE HERE

      REAL*8 RMAN(50), NMAN(50), CHAN(50)
      REAL IC(50,2), ME(50,4)
      INTEGER RP(50,2), PE(50), LP(50,3)
      DIMENSION DN(2), GPE(11,2), CD(10,4), MP(50,2), RN(50,50,4),
1      IPD(50)

      INDEXING AND COUNTING INFO

      READ (2,10) RN, NM, NM, NM, NC, NP, NL, NUQ, NUDE, IFG
10 FORMAT (10(I3,1X))
      DO 20 I = 1, NR
15      MPAT (2,15) RMAN(I), RP(I,1), RP(I,2), IPD(I)
20      CONTINUE

      MESSAGE INFO

      FOUTE INFO

      DO 30 I = 1, NM
1      READ (2,25) NMAN(I), PE(I,1), ME(I,2), MD(I,1), MD(I,2),
25      PD(I,1), MD(I,4)
30      CONTINUE
      FORMAT (A8, 2(I1,13), 4(I1,P9,1))

      CHANNEL INFO

      DO 40 I = 1, NC
35      READ (2,35) CHAN(I), CD(I,1), CD(I,2), CD(I,3), CD(I,4)
40      CONTINUE
      FORMAT (A8, 4(I1,P9,1))

      NUMBER OF TEMPLATES

      DO 50 I = 1, NF
45      MPAT (2,45) PD(I)
50      CONTINUE
      FORMAT (2I, 13)

      ITER INFO

```

&lt;PAGE 1&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:DISPLAY /////

&lt;PAGE 1&gt;

&lt;PAGE 2&gt;

```

1 DATE:02-05-80,14:41 CUSER:IDSH FILE:DISPLAY
DO 60 I = 1, NI
  READ (2,55) LP(I,1), IP(I,2), LF(I,3), LD(I,4), LO(I,5), LD(I,6,1X)
55 FORMAT (1(1,1X), 2(12,6,1X))
60 CONTINUE
C
C      FOURTH MESSAGE AREA
C
  IF (IP(I,2) .EQ. 1) READ (2,65) FLAG
65 FORMAT (1X, A9)
  IF (FLAG .EQ. 1) GO TO 85
  NO OF I = 1, NR
  DO 75 J = 1, NR
    READ (2,70) RM(I,J,1), RM(I,J,2), RM(I,J,3), RM(I,J,4)
70 FORMAT (4(E9.3,1X))
75 CONTINUE
80 CONTINUE
85 CONTINUE
C
C      FISH ACCESS TIMES
C
  READ (2,90) CE(I), CD(2)
90 FORMAT (2(E9.3,1X))
C
C      GENERAL PROCESSOR DATA
C
  DO 95 I = 1, 11
    READ (2,90) GPE(I,1), GDE(I,2)
95 CONTINUE
C
C      END OF READ SECTION
C
  WRITE SYSTEM CONFIGURATION HEADERS
C
  WRITE (6,100)
100 FORMAT ('-', 17X, 50(' '), /, 19X, '*****', 40X, '*****')
105 FORMAT (19X, '*****', 40X, '*****', /, 38X, 50(' '), /)
  WRITE (6,110)
110 FORMAT (19X, '*****', 10X, 'SYSTEM CONFIGURATION', 10X, '*****')
  WRITE (6,105)
  IF (NPEP .EQ. 0) GO TO 160
C
C      CONFIGURATION OF BRIGADES
C
  II = FHC + 1
  JJ = BDEF + MPC
  WRITE (6,115)
115 FORMAT ('0', 50X, '*** MANEUVER BRIGADES ***', /, 22X, 'UNIT',
1 16X, 'COMMUNICATIONS', 13X, 'PROCESSOR', 16X, 'NUMBER',
2 10X, /, 22X, 'NAME', 17X, 'LINK #', 17X, 'TYPE',
3 19X, 'TELETYPE', /)
  NO 135 I = II, JJ
  IF (JPD(I) .EQ. 1) GO TO 125
  WRITE (6,120) RNAM(I), CHAN(I,1,2), PD(I)
120 FORMAT (22X, A8, 15X, A8, 19X, 'TCT', 22X, 12)
  GO TO 135
125 WRITE (6,130) RNAM(I), CHAN(I,1,2), PD(I)
130 FORMAT (22X, A8, 15X, A8, 19X, 'TCS', 22X, 12)
135 CONTINUE

```

&lt;PAGE 2&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:DISPLAY /////

&lt;PAGE 2&gt;

&lt;PAGE 3&gt;

DATE:02-05-80,14:41 CMMER:IOSH FILE:DISFIAY

&lt;PAGE 3&gt;

```

117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C

      CONFIGURATION OF BATTALIONS
      IF (JJ .EQ. 0) GO TO 160
      WRITE (6,140)
180 FORMAT ('0', 49X, '** NAMEVER BATTALIONS **', //, 22X, 'UNIT',
1 18X, 'COMMUNICATIONS', 15X, 'PARANT', 17X, 'PROCESSOR', /,
2 22X, 'NAME', 16X, 'LINK W/ BRIGADE', 14X, 'BRIGADE', 18X,
3 'TYPE', /)
      JJ = JJ + 1
      ISPACE = IP(JJ,3)
      DO 160 I = JJ, NR
      IF (IP(I,3) .NE. ISPACE) WRITE (6,140)
      ISPACE = IP(I,3)
      IF (IP(I,3) .EQ. 1) GO TO 150
      WRITE (6,145) NAME(I), NAME(IP(I,2)), NAME(IP(I,3))
195 FORMAT ('2X, A8, 16X, A8, 10X, A8, 17X, 'TCT')
      GO TO 160
150 WRITE (6,155) NAME(I), NAME(IP(I,2)), NAME(IP(I,3))
155 FORMAT ('22X, A8, 16X, A8, 18X, A8, 17X, 'TCS')
160 CONTINUE

      CONFIGURATION OF OTHER TCS USERS
      IF (MNO .EQ. 0) GO TO 185
      WRITE (6,165)
165 FORMAT ('0', 51X, '** OTHER TCS USERS **', //, 38X, 'UNIT', 14X,
1 18X, 'COMMUNICATIONS', 12X, 'PROCESSOR', /, 30X, 'NAME', 19X,
2 11X, 'TYPE', /)
      DO 185 I = 1, NUC
      IF (IP(I,3) .EQ. 1) GO TO 175
      WRITE (6,170) NAME(I), NAME(IP(I,2))
170 FORMAT ('18X, A8, 11X, A8, 10X, 'TCT')
      GO TO 185
175 WRITE (6,180) NAME(I), NAME(IP(I,2))
180 FORMAT ('18X, A8, 11X, A8, 18X, 'TCS')
185 CONTINUE

      WRITE DATA ABOUT THE COMMUNICATIONS NET
      WRITE (6,190)
      WRITE (6,190)
190 FORMAT ('18X, '*****', 9X, 'COMMUNICATIONS SYSTEM', 10X, '*****')
      WRITE (6,195)
      WRITE (6,195)
195 FORMAT ('0', 29X, 'CHANNEL', 9X, 'NUMBER', 7X, 'MISSION', 9X,
1 18X, 'RISF', 7X, 'UNAVAILABLE', /, 31X, 'NAME', 10X, 'OF LINES',
2 18X, 'RATE', 12X, 'TIME', 9X, 'CAPACITY', /, 58X,
3 'CHARA/SEC', 6X, '(SECONDS)', 8X, '(FRACTION)', /)
      DO 205 I = 1, NC
      ICD = CD(I,4)
      JCC = CD(I,2)
      WRITE (6,200) NAME(I), ICD, JCC, CD(I,1), CD(I,3)
200 FORMAT ('29X, A8, 10X, I3, 11X, I4, 12X, F4.2, 11X, F4.2')
205 CONTINUE

      WRITE DATA ABOUT THE MESSAGES
      WRITE (6,210)
      WRITE (6,210)
210 FORMAT ('18X, '*****', 9X, 'MESSAGES SYSTEM', 10X, '*****')
      WRITE (6,215)
      WRITE (6,215)
215 FORMAT ('0', 29X, 'CHANNEL', 9X, 'NUMBER', 7X, 'MISSION', 9X,
1 18X, 'RISF', 7X, 'UNAVAILABLE', /, 31X, 'NAME', 10X, 'OF LINES',
2 18X, 'RATE', 12X, 'TIME', 9X, 'CAPACITY', /, 58X,
3 'CHARA/SEC', 6X, '(SECONDS)', 8X, '(FRACTION)', /)
      DO 225 I = 1, NC
      ICD = CD(I,4)
      JCC = CD(I,2)
      WRITE (6,220) NAME(I), ICD, JCC, CD(I,1), CD(I,3)
220 FORMAT ('29X, A8, 10X, I3, 11X, I4, 12X, F4.2, 11X, F4.2')
225 CONTINUE

```

&lt;PAGE 3&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:DISPLAY /////

&lt;PAGE 3&gt;



&lt;PAGE 5&gt;

 I DATE:02-05-80,14:41 CMMER:105H FILE:DISPLAY  
 I

&lt;PAGE 5&gt;

ISM

```

233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

&lt;PAGE 5&gt;

 I DATE:02-05-80,14:41 CMMER:105H FILE:DISPLAY  
 I

&lt;PAGE 5&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

&lt;PAGE 5&gt;

&lt;PAGE 6&gt;

```

1 DATE:02-05-80,14:41 CYMPE:IDSN FILE:DISPLAY

```

&lt;PAGE 6&gt;

```

291 DO 315 I = 1, JK
292 EC 310 J = 1, NUM
293   ILM = RM(I,J,1)
294   WRITE (6,305) RMAM(I), MMAM(J), ILM
295   FORMAT (41X, A9, 9X, A9, 11X, 13)
296   CONTINUE
297   WRITE (6,440)
298   CONTINUE
299   IF (JF -EQ. N5) GO TO 315
300   WRITE (6,440)
301   WRITE (6,320)
302
303   C
304   C
305   C
306   C
307   C
308   C
309   C
310   C
311   C
312   C
313   C
314   C
315   C
316   C
317   C
318   C
319   C
320   C
321   C
322   C
323   C
324   C
325   C
326   C
327   C
328   C
329   C
330   C
331   C
332   C
333   C
334   C
335   C
336   C
337   C
338   C
339   C
340   C
341   C
342   C
343   C
344   C
345   C
346   C
347   C
348   C
349   C
350   C
351   C
352   C
353   C
354   C
355   C
356   C
357   C
358   C
359   C
360   C
361   C
362   C
363   C
364   C
365   C
366   C
367   C
368   C
369   C
370   C
371   C
372   C
373   C
374   C
375   C
376   C
377   C
378   C
379   C
380   C
381   C
382   C
383   C
384   C
385   C
386   C
387   C
388   C
389   C
390   C
391   C
392   C
393   C
394   C
395   C
396   C
397   C
398   C
399   C
400   C
401   C
402   C
403   C
404   C
405   C
406   C
407   C
408   C
409   C
410   C
411   C
412   C
413   C
414   C
415   C
416   C
417   C
418   C
419   C
420   C
421   C
422   C
423   C
424   C
425   C
426   C
427   C
428   C
429   C
430   C
431   C
432   C
433   C
434   C
435   C
436   C
437   C
438   C
439   C
440   C
441   C
442   C
443   C
444   C
445   C
446   C
447   C
448   C
449   C
450   C
451   C
452   C
453   C
454   C
455   C
456   C
457   C
458   C
459   C
460   C
461   C
462   C
463   C
464   C
465   C
466   C
467   C
468   C
469   C
470   C
471   C
472   C
473   C
474   C
475   C
476   C
477   C
478   C
479   C
480   C
481   C
482   C
483   C
484   C
485   C
486   C
487   C
488   C
489   C
490   C
491   C
492   C
493   C
494   C
495   C
496   C
497   C
498   C
499   C
500   C
501   C
502   C
503   C
504   C
505   C
506   C
507   C
508   C
509   C
510   C
511   C
512   C
513   C
514   C
515   C
516   C
517   C
518   C
519   C
520   C
521   C
522   C
523   C
524   C
525   C
526   C
527   C
528   C
529   C
530   C
531   C
532   C
533   C
534   C
535   C
536   C
537   C
538   C
539   C
540   C
541   C
542   C
543   C
544   C
545   C
546   C
547   C
548   C
549   C
550   C
551   C
552   C
553   C
554   C
555   C
556   C
557   C
558   C
559   C
560   C
561   C
562   C
563   C
564   C
565   C
566   C
567   C
568   C
569   C
570   C
571   C
572   C
573   C
574   C
575   C
576   C
577   C
578   C
579   C
580   C
581   C
582   C
583   C
584   C
585   C
586   C
587   C
588   C
589   C
590   C
591   C
592   C
593   C
594   C
595   C
596   C
597   C
598   C
599   C
600   C
601   C
602   C
603   C
604   C
605   C
606   C
607   C
608   C
609   C
610   C
611   C
612   C
613   C
614   C
615   C
616   C
617   C
618   C
619   C
620   C
621   C
622   C
623   C
624   C
625   C
626   C
627   C
628   C
629   C
630   C
631   C
632   C
633   C
634   C
635   C
636   C
637   C
638   C
639   C
640   C
641   C
642   C
643   C
644   C
645   C
646   C
647   C
648   C
649   C
650   C
651   C
652   C
653   C
654   C
655   C
656   C
657   C
658   C
659   C
660   C
661   C
662   C
663   C
664   C
665   C
666   C
667   C
668   C
669   C
670   C
671   C
672   C
673   C
674   C
675   C
676   C
677   C
678   C
679   C
680   C
681   C
682   C
683   C
684   C
685   C
686   C
687   C
688   C
689   C
690   C
691   C
692   C
693   C
694   C
695   C
696   C
697   C
698   C
699   C
700   C
701   C
702   C
703   C
704   C
705   C
706   C
707   C
708   C
709   C
710   C
711   C
712   C
713   C
714   C
715   C
716   C
717   C
718   C
719   C
720   C
721   C
722   C
723   C
724   C
725   C
726   C
727   C
728   C
729   C
730   C
731   C
732   C
733   C
734   C
735   C
736   C
737   C
738   C
739   C
740   C
741   C
742   C
743   C
744   C
745   C
746   C
747   C
748   C
749   C
750   C
751   C
752   C
753   C
754   C
755   C
756   C
757   C
758   C
759   C
760   C
761   C
762   C
763   C
764   C
765   C
766   C
767   C
768   C
769   C
770   C
771   C
772   C
773   C
774   C
775   C
776   C
777   C
778   C
779   C
780   C
781   C
782   C
783   C
784   C
785   C
786   C
787   C
788   C
789   C
790   C
791   C
792   C
793   C
794   C
795   C
796   C
797   C
798   C
799   C
800   C
801   C
802   C
803   C
804   C
805   C
806   C
807   C
808   C
809   C
810   C
811   C
812   C
813   C
814   C
815   C
816   C
817   C
818   C
819   C
820   C
821   C
822   C
823   C
824   C
825   C
826   C
827   C
828   C
829   C
830   C
831   C
832   C
833   C
834   C
835   C
836   C
837   C
838   C
839   C
840   C
841   C
842   C
843   C
844   C
845   C
846   C
847   C
848   C
849   C
850   C
851   C
852   C
853   C
854   C
855   C
856   C
857   C
858   C
859   C
860   C
861   C
862   C
863   C
864   C
865   C
866   C
867   C
868   C
869   C
870   C
871   C
872   C
873   C
874   C
875   C
876   C
877   C
878   C
879   C
880   C
881   C
882   C
883   C
884   C
885   C
886   C
887   C
888   C
889   C
890   C
891   C
892   C
893   C
894   C
895   C
896   C
897   C
898   C
899   C
900   C
901   C
902   C
903   C
904   C
905   C
906   C
907   C
908   C
909   C
910   C
911   C
912   C
913   C
914   C
915   C
916   C
917   C
918   C
919   C
920   C
921   C
922   C
923   C
924   C
925   C
926   C
927   C
928   C
929   C
930   C
931   C
932   C
933   C
934   C
935   C
936   C
937   C
938   C
939   C
940   C
941   C
942   C
943   C
944   C
945   C
946   C
947   C
948   C
949   C
950   C
951   C
952   C
953   C
954   C
955   C
956   C
957   C
958   C
959   C
960   C
961   C
962   C
963   C
964   C
965   C
966   C
967   C
968   C
969   C
970   C
971   C
972   C
973   C
974   C
975   C
976   C
977   C
978   C
979   C
980   C
981   C
982   C
983   C
984   C
985   C
986   C
987   C
988   C
989   C
990   C
991   C
992   C
993   C
994   C
995   C
996   C
997   C
998   C
999   C
1000   C

```

&lt;PAGE 6&gt;

```

111111 FILE:DISPLAY

```

```

111111 MAIN PROGRAM

```

&lt;PAGE 6&gt;

[illegible]

&lt;PAGE 8&gt;

807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827

```

815 CONTINUE
  IF (JJ .EQ. NR) GO TO 825
  WRITE (6,440)
  JJ = JJ + 1
  ISPACE = LP(JJ,1)
  DO 825 I = JJ, NR
    IF (LP(I,1) .NE. ISPACE) WRITE (6,440)
    ISPACE = LP(I,1)
    WRITE (6,420) RNAME(I), RNAME(LP(I,1)), LD(I,1), RNAME(LP(I,1)),
      1 NAME(I), LD(I,2)
    820 FORMAT (15X, A6, 7X, A6, 9X, P5.1, 22X, A6, 7X, A6, 9X, P5.1)
    825 CONTINUE
  IF (NR .EQ. 0) GO TO 810
  WRITE (6,440)
  DO 810 I = 1, NHC
    WRITE (6,410) RNAME(I), LD(I,1), RNAME(I), LD(I,2)
  810 CONTINUE
  WRITE (6,415)
  815 FORMAT (11)
  820 FORMAT (11)
  END

```

&lt;PAGE 8&gt;

158

1 DATE:02-05-80,18:41 GENE:IDSH FILE:DISPLAY

241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260

&lt;PAGE 8&gt;

///// FILE:DISPLAY /////

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

&lt;PAGE 8&gt;

DATE: 02-05-80, 14:42 CUMUL: 10501 5:57FF FILE: NO15A

DATE: 02-05-80, 14:42 CUMUL: 10501 5:57FF FILE: NO15A

DATE: 02-05-80, 14:42 CUMUL: 10501 5:57FF FILE: NO15A

>>>> MAIN PROGRAM <<<<

FILE:MODIFY ////

**<PAGE 1>**

```

<PAGE 2>
1 DATE:02-05-80, 18:42 C2REF:IDSN FILE:MODIFY
15N
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79

IF (ANSWR -PG. NO) GO TO 130
CALL ERCC
WRITE(6,60)
CALL PREP(5,'S:',ANSWR,1)
IF (ANSWR -PG. NO) GO TO 180
CALL IDFACT
WRITE(6,70)
CALL PREP(5,'S:',ANSWR,1)
IF (ANSWR -PG. NO) GO TO 150
CALL NOISE
WRITE(6,80)
CALL PREP(5,'S:',ANSWR,1)
IF (ANSWR -PG. YES) GO TO 800
C
C WRITE NEW DATA FILE
C
C CALL WRITEF
C
C WRITE(6,90)
C STOP
C END

```

```

<PAGE 2>
>>>> MAIN PROGRAM <<<<
///// FILE:MODIFY /////
<PAGE 2>

```



&lt;PAGE 4&gt;

1 DATE:02-05-80, 14:42 CMHHR:ICSM FILE:MODIFY

158

90 156 1-1, MC

ICG-CELL, 4)

JCB-CELL, 2)

331 WRITE(6,331)CHAR(1),ICD,JCE,CELL,1),CU(1,1)

156 FORMAT(29E,49,10E,13,11E,14,12E,14,2,11E,14,2)

CONTINUE

C 600 WRITE(6,60)

CALL PRPAB(5,'S',ANSWER,1)

IF (ANSWER .EQ. YES) GO TO 900

PRINT

END

&lt;PAGE 4&gt;

138

139

140

141

142

143

144

145

146

147

148

149

&lt;PAGE 4&gt;

&gt;&gt;&gt;&gt; SUBROUTINE CONSIST &lt;&lt;&lt;&lt;

///// FILE:MODIFY /////

&lt;PAGE 4&gt;

<PAGE 5>

1 DATE:02-05-80,14:42 CMMER:1050 FILE:MODIFY 1

13M

1

2

3

SUBROUTINE CONFIG  
RETURN  
END

<PAGE 5>

150  
151  
152

<PAGE 5>

>>>> SUBROUTINE CONFIG <<<<

////// FILE:MODIFY //

<PAGE 5>

```

<PAGE 6>
1
ISM
1
1 DATE:02-05-80,14:42 CUREE:TDSB FILE:NOCTY
SUBROUTINE LDFACT
C
C ALTFES THE PLPMENTS OF THE ROUTP CROSS MESSAGE DATA ARRAY
C CONCERNING INPUT MESSAGE RATES BY MULTIPLYING ALL OF THEM BY A
C LOAD FACTOR READ BELOW.
C
INTEGER NC,NR,M1,MP,MN,MM,MHQ,MNM,NMEE
REAL*8 PNAME(4),DNAMEF(8,5),HNAME(15),CMAN(10)
REAL*8 RNAN(50),RNAP(50)
REAL IP(50,2),MD(50,4)
INTEGER RP(50,2),RMNR,PB(50),BP(8),DN(8,5),HQ(15)
DIMENSION MCM(4),LINK(3,5),LINC(9),LINCK(15),LP(50,1),
*BD(2),GPD(11,2),CD(10,4),MI(50,2),RM(50,50,4),IPB(50)
COMMON JINT, MC,MR,M1,MP,MN,MM,MHQ,MNM,NMEE
COMMON ECHAN,FNAME,HNAME,CMAN
COMMON FRAM,RNAM
COMMON IC,ND
COMMON FP,RNHR,PD,BD,BR,HQ
COMMON YEN,LINK,LINC,LINCK,IP,
*RD,GEE,CU,MP,RM,IPD
C
10 FORMAT(' ENTER LOAD FACTOR:')
WRITE(6,10)
CALL FPARC(5,'R:',FACTOR)
DO 100 I=1,NR
DO 100 J=1,MNR
RM(I,J,1)=RM(I,J,1)*FACTOR
RETURN
END
100
C
170
171
172
173
174
175
176
177
178
179
180
181
182
183

```

```
<PAGE 6>
      //// "TIP:NOTIFY" ////
      >>>> SUBROUTINE LDFACT <<<<
      <PAGE 6>
```

<PAGE 7>

194  
195  
196

SUBROUTINE MSG  
RETURN  
END

<PAGE 7>

DATE: 02-05-80, 14:42 CMMER: IDSH FILE: MODIFY

ISM

1  
2  
3

<PAGE 7>

////// FILE: MODIFY //

>>>> SUPROUTINE MSG <<<<

<PAGE 7>

[illegible]

&lt;PAGE 9&gt;

&lt;PAGE 9&gt;

15M

1 DATE:02-05-80, 14:42 CMMER:IDSH FILE:MODIFY

```

285      'MODP', 11X, 'HATE', /)
286      IF (NOC.FO.O) GOTO 162
287      .JJ=NHQ*NDOP
288      LJ=NRQ*1
289      DO 162 I=1,JJ
290      WRITE(6,101)NHAM(I),LD(I,1),NHAM(I),LD(I,2)
291      FORMAT(15X,A8,7X,'FEF',11X,P8.6,21X,'PER',12X,A8,6X,P8.6)
292      CONTINUE
293      IF (JJ.FO.NR) GOTO 161
294      WRITE(6,104)
295      .JJ=JJ*1
296      ISPACE=IP(JJ,3)
297      DO 163 I=JJ,NR
298      IF (LP(I,3).NE.ISPACE) WRITE(6,104)
299      ISPACE=I(I,3)
300      WRITE(6,302)NHAM(I),NHAM(LP(I,3)),LD(I,1),NHAM(LP(I,3)),
301      *NHAM(I),I(I,2)
302      *FORMAT(15X,A8,7X,A8,6X,P8.6,21X,A8,7X,A8,6X,P8.6)
303      CONTINUE
304      IF (NHC.FO.O) GOTO 164
305      WRITE(6,304)
306      DO 164 I=1,NHC
307      WRITE(6,301)NHAM(I),LD(I,1),NHAM(I),LD(I,2)
308      CONTINUE
309      WRITE(6,315)
310      *FORMAT(11X)
311      FORMAT(' ')
312      C
313      200      WRITE(6,50)
314      CALL PRPAC(5,'S:',ANSWER,1)
315      IF (ANSWER.FO.YES) GO TO 900
316      RETURN
317      END

```

&lt;PAGE 9&gt;

///// FILE:MODIFY /////

&gt;&gt;&gt;&gt; SUBROUTINE NOISP &lt;&lt;&lt;&lt;

&lt;PAGE 9&gt;

<PAGE 10>

278  
279  
280

SUBROUTINE PROC  
RETURN  
END

<PAGE 10>

ISN

1  
2  
3

DATE: 02-05-80, 14:47 CYNFB: TCSH FILE: MODIFY

<PAGE 12>

////// FILE: MODIFY //

>>>> SUBROUTINE PROC <<<<

<PAGE 10>

[illegible]

**<PAGE 11>**

>>>> SUBROUTINE REAP <<<<

FILE:MOIFY ////

**<PAGE 11>**





&lt;PAGE 2&gt;

DATE: 02-05-80, 14:42 CUSER: IOSU FILE: COMEUF

&lt;PAGE 2&gt;

```

56 NPAC(7,903) = RE(1),RD(2)
57 DO (1) = DO(1)/60000.
58 DO (2) = DO(2)/60000.
59 FORMAT(2('0.1,1X'))
60 DO 10 I=1,11
61 NPAC(7,903) = GPD(1,1),GIF(1,2)
62 GPD(1,1) = GPD(1,1)/60000.
63 GPD(1,2) = GIF(1,2)/60000.
64 CONTINUE
65
66 INITIALIZATION
67
68 LARG=10.0*NO
69 DO 10 I=1,NR
70 EC 104 J=1,3
71 ROUT(1,J)=0.
72 CONTINUE
73 EC 11 J=1,NR
74 RMG(J,I)=0.
75 CONTINUE
76 DO 100 I=1,NR
77 EC 105 J=1,3
78 NO(1,J)=0.
79 CONTINUE
80 DO 101 I=1,NR
81 EC 106 J=1,3
82 ROUT(1,J)=0.
83 CONTINUE
84 DO 102 I=1,NP
85 EC 107 J=1,6
86 ROUT(1,J)=0.
87 CONTINUE
88 DO 103 I=1,NC
89 EC 108 J=1,6
90 CONT(1,J)=0.
91 CONTINUE
92 DO 109 I=1,1
93 EC 109 J=1,5
94 ROUT(1,J)=0.
95 CONTINUE
96 DO 100 I=1,NR
97 EC 109 J=1,5
98 ROUT(1,J)=0.
99 CONTINUE
100
101 INERT AND INITIALIZATION MODULE ENDS
102 TRAFFIC FLOW MODULE BEGINS
103
104 COMMENT THE DATE OF ARRIVAL OF EACH TYPE OF
105 TCS GENERATE MESSAGE TO EACH ROUTE
106
107 DO 12 I=1,NR
108 EC 11 J=1,NR
109 RMG(J,I) = RM(J,I,1)
110
111
112
113

```

&lt;PAGE 2&gt;

///// FILE: COMEUF /////

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

&lt;PAGE 2&gt;

```

<PAGE 3>                                     <PAGE 3>
I DATE:02-05-80,14:42 CHANNEL:IDSH FILE:COMPUTE I
ISM
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

C POINTERS TO THE MESSAGE TYPE OF RESPONSES ON THIS ROUTE
C TO THIS TYPE OF USER MESSAGE
C
C      I1=RP(I,1)
C
C POINTER TO THE MESSAGE TYPE OF RESPONSES ON ALL OTHER ROUTES
C TO THIS TYPE OF USER MESSAGE
C
C      I2=RP(1,2)
C
C INCREMENT THE RATE OF OUTPUTS OF THE APPROPRIATE TYPE ON THE
C INCORPORATING ROUTE
C
C      IF (I1.EQ.(0)) GOTO 17
C      RMO(J,I1)=RMC(J,I1)*RM(J,I,1)*(1.-RM(J,I,1))
C      CCNTIMEP
C
C INCREMENT RATES OF OUTPUTS OF THE APPROPRIATE TYPE ON ALL OTHER ROUTES
C
C      IF (I2.EQ.(0)) GOTO 13
C      TO 14 N=1,NH
C      IF (N.EQ.J) GOTO 14
C      RMO(K,I2)=RMO(K,I2)*RMO(J,I,1)*RM(K,I2,2)*(1.-RM(J,I,1))
C      CCNTIMEP
C      CONTINUE
C      CONTINUE
C
C TRAFFIC FLOW MODULE ENDS
C OPERATING STATISTICS MODULE BEGINS
C
C COMPUTE THE AGGREGATE ARRIVAL RATES FROM THE ARRIVAL RATES
C TO EACH ROUTE
C
C DO 21 I=1,NR
C     TO 22 J=1,RH
C
C POINTERS TO THE LINK(S) ON THE ROUTE
C
C      I1=RP(I,1)
C      I2=RP(1,2)
C
C SET POINTERS TO THE NODES AND CHANNEL OF THE FIRST LINK,
C AND SET THE APPROPRIATE ARRIVAL RATES
C
C      M1=LP(I1,1)
C      C1=LP(I1,2)
C      M3=LP(I1,3)
C      M2=RP+C1
C      JPE=TPE(M1)
C
C COMPUTE THE RATE AT WHICH MESSAGES OF TYPE J TRAVEL TO
C OR FROM USER I
C
C      A=RMO(1,J)*(1.+RN(I,J,3)*RM(I,J,M1))
C
C COMPUTE THE RATE AT WHICH MESSAGES OF TYPE J TRAVEL TO
C OR FROM THE FCC TO USER I
C

```

**<PAGE 1>**

>>>> MAIN PROGRAM <<<<

////// FILE:COMPUTR. //

<PAGE 1>

[illegible]

<PAGE 4>

>>>> MAIN PROGRAM <<<<

//// FILE:COMPUTE ////

**<PAGE 4>**

```

<PAGE 5>
1
1 DATE:02-05-80,14:42 CURPR:IOSH FILE:COMPUTF
1
1 IN BOTH DIRECTIONS ON ALL LINKS OF THE ROUTE
1
132 CALL TRANS(PF1,SN1,LD(1,1),MDJ1,NBLOCK,IRMC)
133 CALL TRANS(PF2,SN2,LD(1,2),MDJ1,NBLOCK,IRMC)
134 IF (I2.FO.(0)) GOTO 24
135 CALL TRANS(PF3,SN3,LD(2,1),MDJ1,NBLOCK,IRMC)
136 CALL TRANS(PF4,SN4,LD(2,2),MDJ1,NBLOCK,IRMC)
137 CONTINUE
1
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

<PAGE 6>
1 DATE: 02-05-80, 10:42  CYNEX: 105P  FILE: CONDUIT
1
2 158
3 165
4 166
5 167
6
7 168
8 169
9 170
10 171
11 172
12 173
13
14 174
15 175
16 176
17
18 177
19 178
20 179
21 180
22 181
23 182
24
25 183
26 184
27
28 185
29 186
30 187
31 188
32 189
33 190
34
35 191
36 192
37 193
38 194
39 195
40 196
41
42 197
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

&lt;PAGE 6&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE: CONDUIT /////

&lt;PAGE 6&gt;

```

<PAGE 7>
1 DATE:02-05-80,14:47 CYNER:IOSH FILE:COMPUTF
1
1SM
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

[illegible]

```

<PAGE 9>
110
C
462 IF (NO(N1,1)-LT.(1.)) GOTO 110
463 FCUT(1,1)=1ARG
464 FCUT(1,2)=1ARG
465 FCUT(1,2)=1ARG
466 CONTINUE
467
C
468 DOCC COMPONENT SUMMARY STATISTICS COMPUTATIONS ARE COMPLETED
469
C
470 CCRUTP CHANNEL SUMMARY STATISTICS
471
C
472 NO 120 I=1,NC
473 NI=NPST
474 IF (NO(N1,2)-EQ.(0.)) GOTO 120
475 NON1=NO(N1,1)*CD(L,1)
476
C
477 EXPECTED DELAY AT CHANNEL (PIE)
478
C
479  $CDUT(L,1) = NO(N1,1) * CD(L,1) / NO(N1,2) + CS(1) * NON1 * (1. + NO(N1,3) * 2) / (2 * NO(N1,2) * CD(L,1) * (1. - NON1) * 2)$ 
480
C
481 EXPECTED QUEUE LENGTH
482
C
483  $CCUT(L,2) = CS(1) * NON1 * (1. + NO(N1,3) * 2) / (2 * (1. - NON1) * 2)$ 
484
C
485 TRAFFIC RATE (MSG/HR)
486
C
487  $CCUT(L,1) = 60 * NO(N1,2)$ 
488
C
489 UTILIZATION
490
C
491  $CCUT(L,4) = NON1$ 
492
C
493 CAPACITY OF CHANNELS (MSG/HR)
494
C
495  $CCUT(L,5) = 1 * CCUT(L,1) / CCUT(L,4)$ 
496
C
497 MAKE ADJUSTMENTS IF UTILIZATION >= 1
498
C
499 IF (NON1-LT.(1.)) GOTO 120
500 FCUT(1,1)=1ARG
501 FCUT(1,2)=1ARG
502 CONTINUE
503
C
504 CHANNEL SUMMARY STATISTICS ARE COMPUTE
505
C
506 COEPTP PROCESSOR SUMMARY STATISTICS
507
C
508 DO 170 I=1,NP
509 IF (NO(I,2)-EQ.(0.)) GOTO 170
510
C
511 EXPECTED DELAY (MIN)
512
C
513  $FCUT(I,1) = NO(I,1) / NO(I,2) + NO(I,1) * NO(I,3) * 2) / (2 * NO(I,2) * (1. - NO(I,1)))$ 
514
C
515 EXPECTED QUEUE LENGTH
516
C
517  $FCUT(I,2) = NO(I,1) * (1. + NO(I,3) * 2) / (2 * (1. - NO(I,1)))$ 
518
C
519 TRAFFIC RATE (MSG/HR)
520
C
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

&lt;PAGE 9&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:COMPUTE /////

&lt;PAGE 9&gt;

```

<PAGE 10>
I DATE:02-05-80,10:42  OWNER:IDSH  FILE:COMPUT  I
15M

283      FCUT(1,1)=60*NO(1,2)
      UTILIZATION
284      FCUT(1,4)=NO(1,1)
      CAPACITY OF PROCESSORS (MSG/HR)
285      FCUT(1,5)=8*POUT(1,3)/PCUT(1,4)
      MAKE ADJUSTMENTS IF UTILIZATION >= 1
      IF (NO(1,3)-1X.(1,1)) GOTO 130
      FCUT(1,1)=1ARG
      FCUT(1,2)=1ARG
      CONTINUE
130
      PROCESSOR SUMMARY STATISTICS ARE COMPLETED
      PCUTP SUMMARY STATISTICS
290      DO 140 I=1,NR
      SET POINTERS
      I1=RF(1,1)
      I2=RF(1,2)
      N1=LF(1,1)
      C1=LF(1,2)
      N2=TP(1,1)
      K2=C1*NF
      IF(MO(N1,2)*NO(N2,2)*NC(N1,2).EQ.(0.1)) GOTO 142
      EFFECTED DELAY (MIN)
      FCUT(1,1)=FCUT(1,1)+DOUT(2,1)*DOUT(3,1)*POUT(N1,1)*POUT(N3,1)*COUT(C1,1)
298      ARRIVAL RATE (MSG/HR)
      CC 141 J=1,NR
      FCUT(1,2)=HRC(1,1)*FCUT(1,2)
      CONTINUE
      FCUT(1,2)=60*HRT(1,2)
      PERFECT WAITING TIME ON RCUTP
      Y=POUT(1,1)-60*(RCUT(1,4)/NO(1,3)+POUT(2,4)/DOUT(2,1)
      Y=DOUT(1,4)/FCUT(3,3)+POUT(N1,4)/POUT(N1,3)+POUT(N1,4)/FCUT(N1,3)
      Y=Y*(C1,4)*COUT(C1,4)/COUT(C1,3)
      BRANCH IF NO SECOND LINK, ELSE SET POINTERS AND ADD CONTRIBUTIONS OF 2ND LINK
      IF (12.EQ.(0)) GOTO 142
      C2=LE(12,2)
      N5=LF(12,1)
      M4=C2*NP
      IF(MO(N4,2)*NO(N5,2).EQ.(0.1)) GOTO 142
304
305
306
307
308
      >>>> MAIN PROGRAM <<<<
      <PAGE 10>

```



&lt;PAGE 12&gt;

616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633

&lt;PAGE 12&gt;

DATE:02-05-80,14:47 CWNES:1050 FILE:COMPUT

15N

336  
337  
338  
339  
340  
341  
342  
343  
344

```

DO 201 I=1,NR
  X=ROUT(I,10)
  IF=0
  PC 202 J=1,NR
    IF (ROUT(J,10)-GT,X) IF=IP+1
    IF (ROUT(J,10)-EQ,X) AND.(J.LF,1) IP=IP+1
  CONTINUE
  PCUT(I,1)=IP
CONTINUE

```

202

201

C

C

970

WRITE ROUTE OUTPUT TABLE

FORMAT('I')

WRITE(9,970)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

\*\*\*\*\* TABLE 1: ROUTE SUMMARY STATISTICS \*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

&lt;PAGE 12&gt;

//// FILE:COMPUTE ////

>>>> MAIN PROGRAM <<<<

&lt;PAGE 12&gt;

381  
382  
383  
384  
385

```

CONTINUE
DO 221 I=1,NR
  X=ROUT(I,10)
  IF=0
  PC 222 J=1,NR

```

220

C

C

C

C

C

C

C

C

C

C

C

C

C

MESSAGE SUMMARY STATISTICS TABLE

RANK MESSAGE TYPES

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

IF (ROUT(I,10)-GT,X) IF=IP+1

IF (ROUT(I,10)-EQ,X) AND.(J.LF,1) IP=IP+1

CONTINUE

PCUT(I,1)=IP

CONTINUE

WRITE ROUTE OUTPUT TABLE

FORMAT('I')

WRITE(9,970)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

IF (ROUT(I,10)-GT,X) IF=IP+1

IF (ROUT(I,10)-EQ,X) AND.(J.LF,1) IP=IP+1

CONTINUE

PCUT(I,1)=IP

CONTINUE

WRITE ROUTE OUTPUT TABLE

FORMAT('I')

WRITE(9,970)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

EXPERCTED TRAFFIC

DELAY RATE

(MIN)

(MSG/HR)

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

IF (ROUT(I,10)-GT,X) IF=IP+1

IF (ROUT(I,10)-EQ,X) AND.(J.LF,1) IP=IP+1

CONTINUE

PCUT(I,1)=IP

CONTINUE

WRITE ROUTE OUTPUT TABLE

FORMAT('I')

WRITE(9,970)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

EXPERCTED TRAFFIC

DELAY RATE

(MIN)

(MSG/HR)

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

RANK

IF (ROUT(I,10)-GT,X) IF=IP+1

IF (ROUT(I,10)-EQ,X) AND.(J.LF,1) IP=IP+1

CONTINUE

PCUT(I,1)=IP

CONTINUE

WRITE ROUTE OUTPUT TABLE

FORMAT('I')

WRITE(9,970)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

WRITE(9,924)

EXPERCTED TRAFFIC

DELAY RATE

(MIN)

(MSG/HR)

RANK

R

```

<PAGE 13>
1 DATE:02-05-80, 14:42  CWHFE:EDSH  FIL:COMPUTE  <PAGE 13>

IF (MOUT(J,1E)-GT.X) IP=IP+1
IF ((MOUT(J,IP)-EQ.X).AND. (J.LE.I)) IP=IP+1
CONTINUE
PORT(I,3)=IP
CONTINUE

WRITE MESSAGE OUTPUT TABLE

WRITE(8,970)
WRITE(8,924)
WRITE(8,924)
WRITE(8,924)
WRITE(8,916)
WRITE(8,917)
WRITE(8,911)
WRITE(8,917)
WRITE(8,916)
FORMAT('
FORMAT('
FORMAT('
WRITE(8,924)
WRITE(8,924)
WRITE(8,912)
FORMAT('
FORMAT('
WRITE(8,1001)
WRITE(8,911)
FORMAT('
FORMAT('
WRITE(8,924)
DO 223 I=1,NH
R=Y
IF (IC.EQ.01) GOTO 224
CC 225 J=1,NH
IF (IFIX(MOUT(J,3))-EQ 1) K=J
CONTINUE
CONTINUE
WRITE(8,915) NHAN(K),MOUT(K,1),MCUT(K,2),MOUT(K,3)
IF (IFIX(1/5)-NE.1/5.) GOTO 223
WRITE(8,924)
CONTINUE
FORMAT(' ',A0,1X,3(6X,F8.3))
GOTO 299

PROCESSOR SUMMARY STATISTICS TABLE
RANK PROCESSORS
CONTINUE
M11=MP-1
DO 231 I=1,M11
R=PORT(I,IP)
IE=0
CC 242 J=1,NE
IF (PORT(J,1D)-GT.X) IF=IF+1
IF ((PORT(J,1E)-EQ.X).AND. (J.LE.I)) IP=IP+1
CONTINUE
PCUT(I,6)=IP
CONTINUE
222
221
C
C
C
931
937
936
932
1001
933
225
224
223
535
C
C
C
C
C
240
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
313
```



```

PAGE 15>
-----
DATE: 02-05-90, 14:42  CDATE: 1050  FILE: COMPTE  1
-----
PAGE 15>

910 WRITE(0,924)
911 WRITE(0,936)
912 WRITE(0,937)
913 WRITE(0,961)
914 WRITE(0,937)
915 WRITE(0,936)
916 FORMAT(
917 WRITE(0,924)
918 WRITE(0,924)
919 WRITE(0,942)
920 WRITE(0,943)
921 WRITE(0,944)
922 FORMAT(' CHANNEL
923 WRITE(0,924)
924 DO 261 I=1,NC
925 K=1
926 IF (IC.FO.(0)) GOTO 264
927 FO 265 J=1,NC
928 IF (IPIX(COUT(J,6))-FO.I) K=J
929 CONTINUE
930 CONTINUE
931 WRITE(0,945) CNA(K),COUT(K,1),COUT(K,2),COUT(K,3),COUT(K,4),COUT(K,5),COUT(K,6)
932 IF (PIOT(I/5).NE.1/5.) GOTO 263
933 WRITE(0,924)
934 CONTINUE
935 GOTO 999
936 C
937 C
938 C
939 C
940 C
941 CONTINUE
942 WRITE(0,970)
943 WRITE(0,924)
944 WRITE(0,924)
945 WRITE(0,924)
946 WRITE(0,957)
947 WRITE(0,958)
948 WRITE(0,951)
949 WRITE(0,958)
950 WRITE(0,957)
951 FORMAT(
952 FORMAT(
953 WRITE(0,924)
954 WRITE(0,924)
955 WRITE(0,924)
956 FORMAT(
957 WRITE(0,1002)
958 WRITE(0,952)
959 WRITE(0,951)
960 WRITE(0,951)
961 FORMAT(' CONESENT
962 WRITE(0,924)
963 WRITE(0,954) COUT(1,1),COUT(1,2),COUT(1,3),COUT(1,4),COUT(1,5)
964 FORMAT(' NDP
965 WRITE(0,955) COUT(2,1),COUT(2,2),COUT(2,3),COUT(2,4),COUT(2,5)
966 FORMAT(' ME CONTINO',X,51F4.3,5X)
967 WRITE(0,956) COUT(3,1),COUT(3,2),COUT(3,3),COUT(3,4),COUT(3,5)

***** TABLE 4: CHANNEL SUMMARY STATISTICS *****
***** TABLE 5: ECC COMPONENT SUMMARY STATISTICS *****
***** TABLE 6: ECC COMPONENT OUTPUT TABLE *****

***** FILE:COMPTE *****
***** MAIN PROGRAM *****
*****
PAGE 15>

```

&lt;PAGE 16&gt;

```

-----
| DATE: 02-05-80, 14:42 CMMES: 10:50 FILE: COMPTIF |
-----

```

&lt;PAGE 16&gt;

```

P68      056      *COMPTIF DE CONTRÔLE 4X, 5 (F8, 3, 5X)
P69      060      WRITE(10, 107) ECUT(44, 1), POUT(M1, 2), ECUT(MP, 3), POUT(MT, 4), POUT(MP, 5)
P70      070      *FORMAT: FFP
P71      071      *44, 5 (F8, 3, 5X)
P72      072      GO TO 295
P73      073      C
P74      074      C
P75      075      C
P76      076      C
P77      077      C
P78      078      C
P79      079      C
P80      080      C
P81      081      C
P82      082      C
P83      083      C
P84      084      C
P85      085      C
P86      086      C
P87      087      C
P88      088      C
P89      089      C
P90      090      C
P91      091      C
P92      092      C
P93      093      C
P94      094      C
P95      095      C
P96      096      C
P97      097      C
P98      098      C
P99      099      C
P100     100      C
P101     101      C
P102     102      C
P103     103      C
P104     104      C
P105     105      C
P106     106      C
P107     107      C
P108     108      C
P109     109      C
P110     110      C
P111     111      C
P112     112      C
P113     113      C
P114     114      C
P115     115      C
P116     116      C
P117     117      C
P118     118      C
P119     119      C
P120     120      C
P121     121      C
P122     122      C
P123     123      C
P124     124      C
P125     125      C
P126     126      C
P127     127      C
P128     128      C
P129     129      C
P130     130      C
P131     131      C
P132     132      C
P133     133      C
P134     134      C
P135     135      C
P136     136      C
P137     137      C
P138     138      C
P139     139      C
P140     140      C
P141     141      C
P142     142      C
P143     143      C
P144     144      C
P145     145      C
P146     146      C
P147     147      C
P148     148      C
P149     149      C
P150     150      C
P151     151      C
P152     152      C
P153     153      C
P154     154      C
P155     155      C
P156     156      C
P157     157      C
P158     158      C
P159     159      C
P160     160      C
P161     161      C
P162     162      C
P163     163      C
P164     164      C
P165     165      C
P166     166      C
P167     167      C
P168     168      C
P169     169      C
P170     170      C
P171     171      C
P172     172      C
P173     173      C
P174     174      C
P175     175      C
P176     176      C
P177     177      C
P178     178      C
P179     179      C
P180     180      C
P181     181      C
P182     182      C
P183     183      C
P184     184      C
P185     185      C
P186     186      C
P187     187      C
P188     188      C
P189     189      C
P190     190      C
P191     191      C
P192     192      C
P193     193      C
P194     194      C
P195     195      C
P196     196      C
P197     197      C
P198     198      C
P199     199      C
P200     200      C
P201     201      C
P202     202      C
P203     203      C
P204     204      C
P205     205      C
P206     206      C
P207     207      C
P208     208      C
P209     209      C
P210     210      C
P211     211      C
P212     212      C
P213     213      C
P214     214      C
P215     215      C
P216     216      C
P217     217      C
P218     218      C
P219     219      C
P220     220      C
P221     221      C
P222     222      C
P223     223      C
P224     224      C
P225     225      C
P226     226      C
P227     227      C
P228     228      C
P229     229      C
P230     230      C
P231     231      C
P232     232      C
P233     233      C
P234     234      C
P235     235      C
P236     236      C
P237     237      C
P238     238      C
P239     239      C
P240     240      C
P241     241      C
P242     242      C
P243     243      C
P244     244      C
P245     245      C
P246     246      C
P247     247      C
P248     248      C
P249     249      C
P250     250      C
P251     251      C
P252     252      C
P253     253      C
P254     254      C
P255     255      C
P256     256      C
P257     257      C
P258     258      C
P259     259      C
P260     260      C
P261     261      C
P262     262      C
P263     263      C
P264     264      C
P265     265      C
P266     266      C
P267     267      C
P268     268      C
P269     269      C
P270     270      C
P271     271      C
P272     272      C
P273     273      C
P274     274      C
P275     275      C
P276     276      C
P277     277      C
P278     278      C
P279     279      C
P280     280      C
P281     281      C
P282     282      C
P283     283      C
P284     284      C
P285     285      C
P286     286      C
P287     287      C
P288     288      C
P289     289      C
P290     290      C
P291     291      C
P292     292      C
P293     293      C
P294     294      C
P295     295      C
P296     296      C
P297     297      C
P298     298      C
P299     299      C
P300     300      C
P301     301      C
P302     302      C
P303     303      C
P304     304      C
P305     305      C
P306     306      C
P307     307      C
P308     308      C
P309     309      C
P310     310      C
P311     311      C
P312     312      C
P313     313      C
P314     314      C
P315     315      C
P316     316      C
P317     317      C
P318     318      C
P319     319      C
P320     320      C
P321     321      C
P322     322      C
P323     323      C
P324     324      C
P325     325      C
P326     326      C
P327     327      C
P328     328      C
P329     329      C
P330     330      C
P331     331      C
P332     332      C
P333     333      C
P334     334      C
P335     335      C
P336     336      C
P337     337      C
P338     338      C
P339     339      C
P340     340      C
P341     341      C
P342     342      C
P343     343      C
P344     344      C
P345     345      C
P346     346      C
P347     347      C
P348     348      C
P349     349      C
P350     350      C
P351     351      C
P352     352      C
P353     353      C
P354     354      C
P355     355      C
P356     356      C
P357     357      C
P358     358      C
P359     359      C
P360     360      C
P361     361      C
P362     362      C
P363     363      C
P364     364      C
P365     365      C
P366     366      C
P367     367      C
P368     368      C
P369     369      C
P370     370      C
P371     371      C
P372     372      C
P373     373      C
P374     374      C
P375     375      C
P376     376      C
P377     377      C
P378     378      C
P379     379      C
P380     380      C
P381     381      C
P382     382      C
P383     383      C
P384     384      C
P385     385      C
P386     386      C
P387     387      C
P388     388      C
P389     389      C
P390     390      C
P391     391      C
P392     392      C
P393     393      C
P394     394      C
P395     395      C
P396     396      C
P397     397      C
P398     398      C
P399     399      C
P400     400      C
P401     401      C
P402     402      C
P403     403      C
P404     404      C
P405     405      C
P406     406      C
P407     407      C
P408     408      C
P409     409      C
P410     410      C
P411     411      C
P412     412      C
P413     413      C
P414     414      C
P415     415      C
P416     416      C
P417     417      C
P418     418      C
P419     419      C
P420     420      C
P421     421      C
P422     422      C
P423     423      C
P424     424      C
P425     425      C
P426     426      C
P427     427      C
P428     428      C
P429     429      C
P430     430      C
P431     431      C
P432     432      C
P433     433      C
P434     434      C
P435     435      C
P436     436      C
P437     437      C
P438     438      C
P439     439      C
P440     440      C
P441     441      C
P442     442      C
P443     443      C
P444     444      C
P445     445      C
P446     446      C
P447     447      C
P448     448      C
P449     449      C
P450     450      C
P451     451      C
P452     452      C
P453     453      C
P454     454      C
P455     455      C
P456     456      C
P457     457      C
P458     458      C
P459     459      C
P460     460      C
P461     461      C
P462     462      C
P463     463      C
P464     464      C
P465     465      C
P466     466      C
P467     467      C
P468     468      C
P469     469      C
P470     470      C
P471     471      C
P472     472      C
P473     473      C
P474     474      C
P475     475      C
P476     476      C
P477     477      C
P478     478      C
P479     479      C
P480     480      C
P481     481      C
P482     482      C
P483     483      C
P484     484      C
P485     485      C
P486     486      C
P487     487      C
P488     488      C
P489     489      C
P490     490      C
P491     491      C
P492     492      C
P493     493      C
P494     494      C
P495     495      C
P496     496      C
P497     497      C
P498     498      C
P499     499      C
P500     500      C
P501     501      C
P502     502      C
P503     503      C
P504     504      C
P505     505      C
P506     506      C
P507     507      C
P508     508      C
P509     509      C
P510     510      C
P511     511      C
P512     512      C
P513     513      C
P514     514      C
P515     515      C
P516     516      C
P517     517      C
P518     518      C
P519     519      C
P520     520      C
P521     521      C
P522     522      C
P523     523      C
P524     524      C
P525     525      C
P526     526      C
P527     527      C
P528     528      C
P529     529      C
P530     530      C
P531     531      C
P532     532      C
P533     533      C
P534     534      C
P535     535      C
P536     536      C
P537     537      C
P538     538      C
P539     539      C
P540     540      C
P541     541      C
P542     542      C
P543     543      C
P544     544      C
P545     545      C
P546     546      C
P547     547      C
P548     548      C
P549     549      C
P550     550      C
P551     551      C
P552     552      C
P553     553      C
P554     554      C
P555     555      C
P556     556      C
P557     557      C
P558     558      C
P559     559      C
P560     560      C
P561     561      C
P562     562      C
P563     563      C
P564     564      C
P565     565      C
P566     566      C
P567     567      C
P568     568      C
P569     569      C
P570     570      C
P571     571      C
P572     572      C
P573     573      C
P574     574      C
P575     575      C
P576     576      C
P577     577      C
P578     578      C
P579     579      C
P580     580      C
P581     581      C
P582     582      C
P583     583      C
P584     584      C
P585     585      C
P586     586      C
P587     587      C
P588     588      C
P589     589      C
P590     590      C
P591     591      C
P592     592      C
P593     593      C
P594     594      C
P595     595      C
P596     596      C
P597     597      C
P598     598      C
P599     599      C
P600     600      C
P601     601      C
P602     602      C
P603     603      C
P604     604      C
P605     605      C
P606     606      C
P607     607      C
P608     608      C
P609     609      C
P610     610      C
P611     611      C
P612     612      C
P613     613      C
P614     614      C
P615     615      C
P616     616      C
P617     617      C
P618     618      C
P619     619      C
P620     620      C
P621     621      C
P622     622      C
P623     623      C
P624     624      C
P625     625      C
P626     626      C
P627     627      C
P628     628      C
P629     629      C
P630     630      C
P631     631      C
P632     632      C
P633     633      C
P634     634      C
P635     635      C
P636     636      C
P637     637      C
P638     638      C
P639     639      C
P640     640      C
P641     641      C
P642     642      C
P643     643      C
P644     644      C
P645     645      C
P646     646      C
P647     647      C
P648     648      C
P649     649      C
P650     650      C
P651     651      C
P652     652      C
P653     653      C
P654     654      C
P655     655      C
P656     656      C
P657     657      C
P658     658      C
P659     659      C
P660     660      C
P661     661      C
P662     662      C
P663     663      C
P664     664      C
P665     665      C
P666     666      C
P667     667      C
P668     668      C
P669     669      C
P670     670      C
P671     671      C
P672     672      C
P673     673      C
P674     674      C
P675     675      C
P676     676      C
P677     677      C
P678     678      C
P679     679      C
P680     680      C
P681     681      C
P682     682      C
P683     683      C
P684     684      C
P685     685      C
P686     686      C
P687     687      C
P688     688      C
P689     689      C
P690     690      C
P691     691      C
P692     692      C
P693     693      C
P694     694      C
P695     695      C
P696     696      C
P697     697      C
P698     698      C
P699     699      C
P700     700      C
P701     701      C
P702     702      C
P703     703      C
P704     704      C
P705     705      C
P706     706      C
P707     707      C
P708     708      C
P709     709      C
P710     710      C
P711     711      C
P712     712      C
P713     713      C
P714     714      C
P715     715      C
P716     716      C
P717     717      C
P718     718      C
P719     719      C
P720     720      C
P721     721      C
P722     722      C
P723     723      C
P724     724      C
P725     725      C
P726     726      C
P727     727      C
P728     728      C
P729     729      C
P730     730      C
P731     731      C
P732     732      C
P733     733      C
P734     734      C
P735     735      C
P736     736      C
P737     737      C
P738     738      C
P739     739      C
P740     740      C
P741     741      C
P742     742      C
P743     743      C
P744     744      C
P745     745      C
P746     746      C
P747     747      C
P748     748      C
P749     749      C
P750     750      C
P751     751      C
P752     752      C
P753     753      C
P754     754      C
P755     755      C
P756     756      C
P757     757      C
P758     758      C
P759     759      C
P760     760      C
P761     761      C
P762     762      C
P763     763      C
P764     764      C
P765     765      C
P766     766      C
P767     767      C
P768     768      C
P769     769      C
P770     770      C
P771     771      C
P772     772      C
P773     773      C
P774     774      C
P775     775      C
P776     776      C
P777     777      C
P778     778      C
P779     779      C
P780     780      C
P781     781      C
P782     782      C
P783     783      C
P784     784      C
P785     785      C
P786     786      C
P787     787      C
P788     788      C
P789     789      C
P790     790      C
P791     791      C
P792     792      C
P793     793      C
P794     794      C
P795     795      C
P796     796      C
P797     797      C
P798     798      C
P799     799      C
P800     800      C
P801     801      C
P802     802      C
P803     803      C
P804     804      C
P805     805      C
P806     806      C
P807     807      C
P808     808      C
P809     809      C
P810     810      C
P811     811      C
P812     812      C
P813     813      C
P814     814      C
P815     815      C
P816     816      C
P817     817      C
P818     818      C
P819     819      C
P820     820      C
P821     821      C
P822     822      C
P823     823      C
P824     824      C
P825     825      C
P826     826      C
P827     827      C
P828     828      C
P829     829      C
P830     830      C
P831     831      C
P832     832      C
P833     833      C
P834     834      C
P835     835      C
P836     836      C
P837     837      C
P838     838      C
P839     839      C
P840     840      C
P841     841      C
P842     842      C
P843     843      C
P844     844      C
P845     845      C
P846     846      C
P847     847      C
P848     848      C
P849     849      C
P850     850      C
P851     851      C
P852     852      C
P853     853      C
P854     854      C
P855     855      C
P856     856      C
P857     857      C
P858     858      C
P859     859      C
P860     860      C
P861     861      C
P862     862      C
P863     863      C
P864     864      C
P865     865      C
P866     866      C
P867     867      C
P868     868      C
P869     869      C
P870     870      C
P871     871      C
P872     872      C
P873     873      C
P874     874      C
P875     875      C
P876     876      C
P877     877      C
P878     878      C
P879     879      C
P880     880      C
P881     881      C
P882     882      C
P883     883      C
P884     884      C
P885     885      C
P886     886      C
P887     887      C
P888     888      C
P889     889      C
P890     890      C
P891     891      C
P892     892      C
P893     893      C
P894     894      C
P895     895      C
P896     896      C
P897     897      C
P898     898      C
P899     899      C
P900     900      C
P901     901      C
P902     902      C
P903     903      C
P904     904      C
P905     905      C
P906     906      C
P907     907      C
P908     908      C
P909     909      C
P910     910      C
P911     911      C
P912     912      C
P913     913      C
P914     914      C
P915     915      C
P916     916      C
P917     917      C
P918     918      C
P919     919      C
P920     920      C
P921     921      C
P922     922      C
P923     923      C
P924     924      C
P925     925      C
P926     926      C
P927     927      C
P928     928      C
P929     929      C
P930     930      C
P931     931      C
P932     932      C
P933     933      C
P934     934      C
P935     935      C
P936     936      C
P937     937      C
P938     938      C
P939     939      C
P940     940      C
P941     941      C
P942     942      C
P943     943      C
P944     944      C
P945     945      C
P946     946      C
P947     947      C
P948     948      C
P949     949      C
P950     950      C
P951     951      C
P952     952      C
P953     953      C
P954     954      C
P955     955      C
P956     956      C
P957     957      C
P958     958      C
P959     959      C
P960     960      C
P961     961      C
P962     962      C
P963     963      C
P964     964      C
P965     965      C
P966     966      C
P967     967      C
P968     968      C
P969     969      C
P970     970      C
P971     971      C
P972     972      C
P973     973      C
P974     974      C
P975     975      C
P976     976      C
P977     977      C
P978     978      C
P979     979      C
P980     980      C
P981     981      C
P982     982      C
P983     983      C
P984     984      C
P985     985      C
P986     986      C
P987     987      C
P988     988      C
P989     989      C
P990     990      C
P991     991      C
P992     992      C
P993     993      C
P994     994      C
P995     995      C
P996     996      C
P997     997      C
P998     998      C
P999     999      C
P1000    1000     C

```

&lt;PAGE 16&gt;

&gt;&gt;&gt;&gt; MAIN PROGRAM &lt;&lt;&lt;&lt;

///// FILE:COMPUTE. /////

&lt;PAGE 16&gt;

&lt;PAGE 17&gt;

1 DATE: 02-05-80, 14:42 CMMER: 105H FILE: COMPTF 1

&lt;PAGE 17&gt;

ISM

```

878 C
879 C SUBROUTINE TO COMPUTE THE FIRST AND SECOND MOMENTS OF THE
880 C DISTRIBUTION OF THE NUMBER OF TRANSMISSIONS TO SUCCESS.
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C

```

FM = FIRST MOMENT  
 SM = SECOND MOMENT  
 EQ = ERROR RATE AFTER MAJORITY AND BEFORE HAMMING CODE  
 L = NUMBER OF CHARACTERS IN THE MESSAGE  
 NB = PICKING NUMBER  
 TRAC = FRC SWITCH, 0=>ON  
 SCRATCH PAD  
 PRCM = THE PROBABILITY THAT ALL COPIES OF A GIVEN CHARACTER  
 IN A TRANSMISSION (NE CHARACTERS) EACH HAVE 2 OR MORE  
 BITS IN ERROR  
 FRC = FRC SWITCH, 0 OR 1  
 EN = COUNTER  
 PGLTN = PROBABILITY OF SUCCESS IN LESS THAN N TRIES  
 PGLFM = PROBABILITY OF SUCCESS IN LESS THAN OR EQUAL TO N TRIES  
 PACK = PROBABILITY OF A ACK TO A TRANSMISSION OF A MESSAGE

GW ADI-3 1/7/79

```

SUBROUTINE TRANS(FM,SM,PR,L,RD,TRAC)
  PRCM=(1-((1-FR)**12+12*FR*(1-FR)**11))*ND
  PACK=(1-PRCM)**L
  FM=10.**10
  IF (EACH.CH.(10.**(-10))) FM=1/PACK
  SM=2**FM*FM
  IF (IFRC.EQ.(0)).AND.(FM.LT.(10.**10))) GOTO 200
  RTURN
  EN=1.0
  PGLTN=0.0
  PGLFM=0.0
  FM=0.0
  SM=0.0
  CONTINUE
  PGLER=(1-PRCM**EN)**L
  FM=FM+FM*(PGLFM-PGLTN)
  SM=SM+FM*(PGLFM-PGLTN)
  PGLTN=PGLER
  FM=FM+1
  IF (EGLM.LT.(0.999)) GOTO 100
  RETURN
END

```

200  
 100

&lt;PAGE 17&gt;

///// FILE:COMPTF /////

&gt;&gt;&gt;&gt; SUBROUTINE TRANS &lt;&lt;&lt;&lt;

&lt;PAGE 17&gt;

## APPENDIX B: MERGED VARIABLE CROSS-REFERENCE LISTING

This appendix contains a merged variable cross-reference listing for all of the programs which make up the TOS Simulation Package. These programs are CREATE, DISPLAY, MODIFY, and COMPUTE. The merged cross-reference lists all the occurrences of each variable in the programs which make up the TOS Simulation Package.

Referencing of statements that a variable is used in is done using internal statement numbers (see the discussion of internal statement numbers in the previous description of appendix A.) Since internal statements are numbered beginning at one within each routine, each reference to internal statement numbers also requires specification of the routine in which the statement is located. This is accomplished by first listing the variable being cross-referenced, then the routine in which occurrences of this variable appear, and to the far right, a list of internal statement numbers referring to statements in this program module at which the variable is referenced. If the variable is referenced multiple times within a single statement, only one reference appears in the cross-reference listing. Some internal statement references in the listing may be immediately followed by a symbol indicating the usage of the variable in this statement. The following key defines the usage of these symbols:

- \* A variable or a function is changed either through an assignment READ, ASSIGN statement, or is used as a DO index.
- ? A variable may be changed because it is used as a simple argument to a subroutine or function.
- D A subprogram is defined by the SUBROUTINE, FUNCTION, ENTRY, or EXTERNAL statement. A statement function is defined by the statement function definition. A variable is declared in a

type or DIMENSION statement. For units, "D" stands for DEFINE FILE statements.

E A variable appears in an EQUIVALENCE statement.

C A variable appears in a COMMON statement.

These suffixes are especially useful since they allow distinguishing the use of variables in declarations from their use in executable statements, and, in the latter case, the distinguishing between just the referencing of a variable versus its updating. These distinctions can be made, for the most part, just from the suffixes, without the need for looking up the statement in the program listing.

Certain other useful pieces of information are also provided in these cross-reference listings. The type of each variable is indicated. The list of variable types included are "I\*4" for integer variables, "R\*4" for real variables, "L\*4" for logical variables, and "CHAR" for character variables. The field in the cross-reference listing with the heading "ATTR" distinguishes between scalars and arrays. This field is left blank for scalars while it contains "ARRAY" for arrays. It has been found useful to include references to subroutines and functions in these cross-reference listings, in addition to references to just variables. These are distinguished from variables by having "SUBR" or "FCN" in the ATTR field. Another type of item included in these cross-references are RETURN or STOP statements within each program module. These are referenced in the listing using <EXIT> as the name of the "variable".

[illegible]



[illegible]

IA	COMPUTE (1.00)	1332	334	345	342	384	387	388	429	432	433	473	476
IB	COMPUTE (1.00)	1332	338	341									
IC	COMPUTE (1.00)	477	332	415	460	500							
IC	DISP (1.00)	121	48										
IC	COMES (1.00)	46*	62										
IC	DISP (1.00)	95*	97										
IC	CREATE (1.00)	1302	331	333	6227	677	699	701					
IC	CREATE (1.00)	396	397	398	399	400	401	402	405	407	408	409	410
IC	CREATE (1.00)	413	414*	415	644*	649*	652	655					
IC	DISP (1.00)	1	11*	121									
IC	DISP (1.00)	6	28	30	148	150							
IC	COMPUTE (1.00)	239	171	417	462	502							
IC	DISP (1.00)	13*	17	88*	89	173*	175						
IC	DISP (1.00)	50*	54	122*	123	200*	202						
IC	CREATE (1.00)	34*	35	77*	78	111*	112	203*	204	647*	648		
IC	CREATE (1.00)	248*	249	276*	299*	300	432*	433					
IC	DISP (1.00)	169*	176	190	199	213*	214						
IC	DISP (1.00)	196*	201	216	224	238*	239						
IC	DISP (1.00)	48*	49										
IC	DISP (1.00)	71*	76	78	80	82	90*	93					
IC	DISP (1.00)	100*	111	113	115	117	124*	127					
IC	DISP (1.00)	78*	76	78	80	82	91*	93					
IC	DISP (1.00)	109*	111	113	115	117	125*	127					
IC	DISP (1.00)	75*	76	78	80	82	92*	93					
IC	DISP (1.00)	110*	111	113	115	117	126*	127					
IC	CREATE (1.00)	20	2612	2642	2672	2682	1082	1092	1102				
IC	CREATE (1.00)	139*	141*	142*	344	385*	387*	390					
IC	COMPUTE (1.00)	478*	476*	477*	479								
IC	COMPUTE (1.00)	40	11*	106									
IC	CREATE (1.00)	60	105*	198*	212*	218	219*	220*	225	226	6727	680	
IC	DISP (1.00)	1	50	18	33	45							
IC	DISP (1.00)	50*	54	55	70	91							
IC	READ (1.00)	70	130	17*									
IC	WRITE (1.00)	70	130	17									
IC	DISP (1.00)	133*	134	149*	150	162*	163						
IC	DISP (1.00)	162*	163	178*	179	190*	191						
IC	COMPUTE (1.00)	77	1312	1342	1362	1372							
IC	TRANS (1.00)	1	7										
IC	DISP (1.00)	29*	31	12*	144*	146	147*	186*	188	189*	224	225*	225*
IC	DISP (1.00)	66*	68	69*	173*	175	176*	212*	214	215*	246*	248	249*
IC	DISP (1.00)	56*	58	59*									
IC	CREATE (1.00)	218*	220	225*	456	6727							
IC	DISP (1.00)	222*	225*	227	400	522	6727						
IC	DISP (1.00)	1	107	226*	227	400	522	6727					
IC	DISP (1.00)	1	111										
IC	CREATE (1.00)	107	11	29	92	57	67	68	241	242	252	269	273
IC	CREATE (1.00)	314	460	484	500	550	570	592	622	641			
IC	CREATE (1.00)	6712	672										
IC	CREATE (1.00)	6757	676	89									
IC	CREATE (1.00)	86*	88										
IC	CREATE (1.00)	189*	191	94									
IC	CREATE (1.00)	87*	91	94									
IC	CREATE (1.00)	190*	191	200									
IC	CREATE (1.00)	34*	35*	36	51*	52	54*	55	59*	60	64*	65	69*
IC	COMPUTE (1.00)	70	74*	75	79*	80	84*	85	89	93	94	99*	107
IC	COMPUTE (1.00)	108	114	119	144	146	147	152	154	155	165	167	168



[illegible]







## APPENDIX C: STATEMENT LABEL REFERENCES

This Appendix presents statement label reference listings for each program in the analysis package. The statement labels are expressed in terms of internal statement numbers. The program listings in Appendix A can be consulted to look up any referenced statement. Within each program module, for each FORTRAN statement label which appears, four types of information are provided in the label reference listing: the label being referenced; the internal statement number where it is defined (i.e., the statement in which it is a label); a type indicator ("FMT" if it refers to a format statement, and blank otherwise); and a list of statement in the routine in which this statement label is referenced.

## \*\*\* IADPL DICTIONARY -- CREATE \*\*\*

INDEX	DEPN	TYPE	REFERENCES
10	9	PMT	8
11	12	PMT	11
13	14	PMT	13
19	20	PMT	19
22	23	PMT	22
39	27	PMT	39
50	30	PMT	50
103	31	PMT	103
36	37	PMT	36
35	43	PMT	35
33	44	PMT	33
119	45	PMT	119
46	47	PMT	46
51	52	PMT	51
50	58	PMT	50
127	55	PMT	127
60	61	PMT	60
64	65	PMT	64
63	69	PMT	63
75	76	PMT	75
74	79	PMT	74
71	80	PMT	71
72	93	PMT	72
136	86	PMT	136
81	87	PMT	81
70	91	PMT	70
160	94	PMT	160
89	96	PMT	89
29	99	PMT	29
98	102	PMT	98
97	104	PMT	97
42	114	PMT	42
113	117	PMT	113
112	118	PMT	112
109	120	PMT	109
57	122	PMT	57
121	128	PMT	121
67	131	PMT	67
129	133	PMT	129
68	134	PMT	68
163	142	PMT	163
140	145	PMT	140
139	146	PMT	139
151	152	PMT	151
150	155	PMT	150
147	156	PMT	147
148	157	PMT	148
135	165	PMT	135
162	166	PMT	162
95	167	PMT	95
158	173	PMT	158
159	174	PMT	159
88	178	PMT	88
170	182	PMT	170
169	186	PMT	169
178	190	PMT	178

287	187		177	
285	196		191	
290	212		204	
295	218		201	202
303	215		198	
305	221		217	
310	226		224	
315	229		227	
320	232	VMT	231	
325	235	VMT	236	
331	241		237	
335	244	FMT	236	
340	250		249	
345	253		247	
353	254		245	289 303
355	259	FMT	258	296 314
360	262	FMT	261	
365	266	FMT	265	
370	270		257	
375	273		256	
383	281		275	
385	283		241	
390	286	FMT	245	300
395	288		244	
403	290		242	
405	292	FMT	291	
410	295		252	
415	302		298	
423	304		269	
425	307	FMT	306	
430	311		305	
435	317		273	
443	322	FMT	321	
445	326		320	
450	327		272	312 318
455	329	FMT	328	
460	332	FMT	311	
465	340	FMT	319	365
473	343	FMT	346	397
475	351	FMT	350	400
480	355		318	
485	356	FMT	355	427
490	360		317	354
495	361		316	
500	363		314	
505	373	FMT	372	384 414 401
513	374		371	
515	375		370	
523	385		383	
525	390		382	
533	391		391	417
535	406	FMT	405	
540	411		404	
545	412		403	
550	422		416	
555	423		415	
563	424		392 413	
565	434		413	
573	439		411	

535	447				
543	446				
545	449				
590	451	FMT			
595	454	FMT			
600	456	FMT			
605	462	FMT			
610	465	FMT			
615	468	FMT			
620	471	FMT			
625	474				
630	478				
635	475				
640	482	FMT			
645	486	FMT			
651	489	FMT			
655	492	FMT			
660	495				
665	495				
671	501				
675	503	FMT			
680	506	FMT			
685	509				
693	511	FMT			
695	514	FMT			
700	517	FMT			
705	520	FMT			
710	524	FMT			
715	526				
721	532				
725	535	FMT			
731	539				
735	542	FMT			
740	545	FMT			
745	548	FMT			
751	551				
755	553				
760	556	FMT			
765	558				
770	560				
775	562	FMT			
780	565	FMT			
785	568	FMT			
790	571				
795	572				
800	576	FMT			
805	578				
810	579				
815	582				
820	584				
825	590	FMT			
830	593				
835	594				
840	595	FMT			
845	598				
850	603	FMT			
855	605				
860	606				
865	608	FMT			
442	441				
443	443				
450	450				
453	453				
457	457				
461	461				
464	464				
467	467				
470	470				
476	476				
475	475				
473	473				
481	481				
485	485				
488	488				
491	491				
490	490				
496	496				
494	494				
502	502				
505	505				
460	460				
510	510				
511	511				
516	516				
519	519				
523	523				
494	494				
500	500				
524	524				
508	508				
511	511				
544	544				
547	547				
580	580				
550	550				
555	555				
554	554				
552	552				
561	561				
564	564				
567	567				
560	560				
570	570				
575	575				
574	574				
572	572				
581	581				
580	580				
589	589				
588	588				
585	585				
594	594				
592	592				
602	602				
601	601				
599	599				
607	607				
362	425	400			
522	527	517			
529	531				

070	610		597
075	615		611
080	616		611
085	615	PMT	618
090	621		617
095	625		622
000	628	PMT	627
005	641		626
010	642		628
015	646	PMT	635
020	645	PMT	638
025	642		644
030	651	PMT	650
035	654	PMT	653
040	656		648
045	657		695
050	655		641
055	662	PMT	661
060	665	PMT	664
065	667		660
070	668		641
075	670	PMT	669
080	674	PMT	673
085	674	PMT	677
090	681	PMT	680
095	682		679
100	685	PMT	684
105	686		683
110	689	PMT	688
115	690		687
120	692	PMT	692
125	694		691
130	697	PMT	696
135	698		695
140	700	PMT	699
145	705	PMT	704
150	706		701
155	707		702
160	708		701
165	710	PMT	709
170	713		711
175	714		676

## \*\*\* LABEL DICTIONARY -- CHECK \*\*\*

LABEL	OPN	TYPE	REFERENCES
10	3		9
15	7	PMT	6

## \*\*\* LABEL DICTIONARY -- CHECK1 \*\*\*

LABEL	OPN	TYPE	REFERENCES
10	3		12
15	8		4
20	10	PMT	9

## \*\*\* LABEL DICTIONARY -- CHECK2 \*\*\*

LABEL	DEFN	TYPE	REFERENCES
13	4		14
15	10		6
20	12	FMT	7
25	15		5

## \*\*\* LABEL DICTIONARY -- CHECK3 \*\*\*

LABEL	DEFN	TYPE	REFERENCES
10	2		7
15	4	FMT	7

## \*\*\* LABEL DICTIONARY -- CHECK4 \*\*\*

LABEL	DEFN	TYPE	REFERENCES
13	2		7
15	4	FMT	3

## \*\*\* LABEL DICTIONARY -- CLSP \*\*\*

LABEL	DEFN	TYPE	REFERENCES						
13	7	FMT	6	53	65	97	115	205	
15	8	FMT	11	56	68	100	118	208	
20	10	FMT	9						
25	10	FMT	15						
30	20	FMT	14						
35	22	FMT	18						
40	23	FMT	22						
45	24	FMT	17	21					
50	27	FMT	26						
55	35	FMT	34						
60	37	FMT	31						
65	38	FMT	37						
70	35		30	36					
75	40		12	25					
80	43	FMT	42						
85	47	FMT	46						
90	45		45						
95	50	FMT	49						
100	51		44	40					
105	52		41						
110	55	FMT	54						
115	58	FMT	57						
120	63	FMT	62						
125	64		59						
130	67	FMT	66						
135	70	FMT	69						
140	77	FMT	76						
145	74	FMT	74						
150	81	FMT	80						

155	P3	PMT	112	163	
163	P4		111		
165	P5		101		
171	P6	PMT	100		
175	P7		108		
180	P8		105		
185	P9	PMT	104		
190	P10	PMT	103		
195	P11	PMT	102		
200	P12	PMT	101		
205	P13	PMT	100		
210	P14	PMT	108		
215	P15	PMT	107		
220	P16	PMT	112		
225	P17	PMT	111		
230	P18	PMT	116		
235	P19	PMT	119		
240	P20	PMT	124		
245	P21	PMT	127		
			114	163	
250	P22		112		
255	P23		111		
260	P24	PMT	101		
265	P25	PMT	100		
270	P26		108		
275	P27		105		
280	P28		104		
285	P29	PMT	103		
290	P30	PMT	102		
295	P31	PMT	101		
300	P32	PMT	100		
305	P33	PMT	108		
310	P34	PMT	107		
315	P35	PMT	112		
320	P36	PMT	111		
325	P37		116		
330	P38		119		
335	P39		124		
340	P40		127		
345	P41		114	163	
350	P42		112		
355	P43		111		
360	P44	PMT	101		
365	P45	PMT	100		
370	P46	PMT	108		
375	P47	PMT	105		
380	P48	PMT	104		
385	P49	PMT	103		
390	P50	PMT	102		
395	P51	PMT	101		
400	P52	PMT	100		
405	P53	PMT	108		
410	P54	PMT	107		
415	P55	PMT	112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		
			102		
			101		
			100		
			108		
			107		
			112		
			111		
			116		
			119		
			124		
			127		
			114	163	
			112		
			111		
			101		
			100		
			108		
			105		
			104		
			103		

## \*\*\* LABEL DICTIONARY -- DISPLAY \*\*\*

LABEL	WPM	TYPE	REFERENCES
13	7	FMT	6
15	10	FMT	9
20	11		8
25	14	FMT	13
30	15		12
35	18	FMT	17
40	19		16
45	22	FMT	21
50	23		20
55	26	FMT	25
60	27		24
65	29	FMT	28
70	34	FMT	33
75	35		32
80	36		31
85	37		30
90	39	FMT	41
95	42		40
100	44	FMT	43
105	45	FMT	48
110	47	FMT	46
115	53	FMT	52
120	57	FMT	56
125	59		55
130	60	FMT	59
135	61		58
140	64	FMT	63
145	72	FMT	71
150	74		70
155	75	FMT	74
160	76		62
165	79	FMT	67
170	83	FMT	73
175	85		82
180	86	FMT	81
185	87		85
190	90	FMT	77
195	93	FMT	89
200	98	FMT	92
205	99		97
210	102	FMT	94
215	105	FMT	101
220	112	FMT	104
225	114	FMT	111
230	116	FMT	113
235	118		115
240	119	FMT	117
245	124	FMT	107
250	129		127
			120
			123
255	132	FMT	131
260	135	FMT	134
265	137	FMT	136
270	139	FMT	138
275	141	FMT	140
			144
			147
			149
			230
			233



## \*\*\* LABEL DICTIONARY -- MODITY \*\*\*

LABEL	DEFN	TYPE	REFERENCES
10	2	PMT	25
20	3	PMT	26
30	4	PMT	30
40	5	PMT	34
50	6	PMT	38
60	7	PMT	42
70	8	PMT	46
80	9	PMT	50
90	10	PMT	54
100	10	PMT	28
110	14		32
120	16		16
130	42		40
140	46		44
150	50		48
160	26		52

## \*\*\* LABEL DICTIONARY -- CONSYS \*\*\*

LABEL	DEFN	TYPE	REFERENCES
10	16	PMT	23
20	17	PMT	26
30	18	PMT	28
40	15	PMT	29
50	20	PMT	38
60	21	PMT	51
70	22	PMT	11
80	50	PMT	45
90	18	PMT	42
100	17	PMT	19
110	41	PMT	40
120	44	PMT	43
130	49	PMT	48
140	33		30
150	51		36
160	51		36
170	23		53

## \*\*\* LABEL DICTIONARY -- IMPACT \*\*\*

LABEL	DEFN	TYPE	REFERENCES
10	14	PMT	15
100	19		17
			10

## \*\*\* LABEL DICTIONARY -- NOISE \*\*\*

LABEL	DEFN	TYPE	REFERENCES
10	16	PMT	21

20	17	FMT	24
35	18	FMT	27
40	19	FMT	35
50	20	FMT	31
60	21		29
70	22		26
80	23		46
90	24		53
	25		57
	26		65
	27		
	28		
	29		
	30		
	31		
	32		
	33		
	34		
	35		
	36		
	37		
	38		
	39		
	40		
	41		
	42		
	43		
	44		
	45		
	46		
	47		
	48		
	49		
	50		
	51		
	52		
	53		
	54		
	55		
	56		
	57		
	58		
	59		
	60		
	61		
	62		
	63		
	64		
	65		
	66		
	67		
	68		
	69		
	70		
	71		
	72		
	73		
	74		
	75		
	76		
	77		
	78		
	79		
	80		
	81		
	82		
	83		
	84		
	85		
	86		
	87		
	88		
	89		
	90		
	91		
	92		
	93		
	94		
	95		
	96		
	97		
	98		
	99		
	100		

## \*\*\* IAPPL DICTIONARY -- READP \*\*\*

IAPPL	TYPE	REFERENCES
27	18	16
28	22	20
29	26	24
30	30	28
31	34	32
32	38	36
33	42	40
34	46	44
35	50	48
36	54	52
37	58	56
38	62	60
39	66	64
40	70	68
41	74	72
42	78	76
43	82	80
44	86	84
45	90	88
46	94	92
47	98	96
48	102	100
49	106	104
50	110	108
51	114	112
52	118	116
53	122	120
54	126	124
55	130	128
56	134	132
57	138	136
58	142	140
59	146	144
60	150	148
61	154	152
62	158	156
63	162	160
64	166	164
65	170	168
66	174	172
67	178	176
68	182	180
69	186	184
70	190	188
71	194	192
72	198	196
73	202	200
74	206	204
75	210	208
76	214	212
77	218	216
78	222	220
79	226	224
80	230	228
81	234	232
82	238	236
83	242	240
84	246	244
85	250	248
86	254	252
87	258	256
88	262	260
89	266	264
90	270	268
91	274	272
92	278	276
93	282	280
94	286	284
95	290	288
96	294	292
97	298	296
98	302	300
99	306	304
100	310	308

## \*\*\* IAPPL DICTIONARY -- WRITE \*\*\*

IAPPL	TYPE	REFERENCES
27	18	16
28	22	20
29	26	24
30	30	28
31	34	32
32	38	36
33	42	40
34	46	44
35	50	48
36	54	52
37	58	56
38	62	60
39	66	64
40	70	68
41	74	72
42	78	76
43	82	80
44	86	84
45	90	88
46	94	92
47	98	96
48	102	100
49	106	104
50	110	108
51	114	112
52	118	116
53	122	120
54	126	124
55	130	128
56	134	132
57	138	136
58	142	140
59	146	144
60	150	148
61	154	152
62	158	156
63	162	160
64	166	164
65	170	168
66	174	172
67	178	176
68	182	180
69	186	184
70	190	188
71	194	192
72	198	196
73	202	200
74	206	204
75	210	208
76	214	212
77	218	216
78	222	220
79	226	224
80	230	228
81	234	232
82	238	236
83	242	240
84	246	244
85	250	248
86	254	252
87	258	256
88	262	260
89	266	264
90	270	268
91	274	272
92	278	276
93	282	280
94	286	284
95	290	288
96	294	292
97	298	296
98	302	300
99	306	304
100	310	308

903	27	FMT	25
904	31	FMT	29
905	35	FMT	33
906	41	FMT	38
907	43	FMT	42
			45

## \*\*\* LABEL DICTIONARY -- COMPUTE \*\*\*

LABEL	DEFN	TYPE	REFERENCES	
			CO	
10	57		50	
11	56		54	
12	97		63	
13	56		84	91
14	95		92	93
17	90		88	
21	235		98	
22	234		99	
23	112		109	
24	118		115	
25	185		139	
26	177		148	
27	224		144	
28	217		200	
29	248		236	
33	12		10	
34	17		14	
35	23		19	
36	27		25	
37	48		44	
38	11		29	
39	16		33	
40	17		34	
41	62		58	
42	67		63	
43	72		68	
44	77		73	
45	53		51	
46	61		59	
47	66		64	
48	71		69	
49	76		74	
50	82		78	
51	266		255	263
52	254		249	250
53	275		267	269
54	245		242	276
55	290		280	287
56	316		291	
57	302		300	
58	312		298	305
59	315		313	309
60	321		317	
61	325		318	
62	326		322	323
63	31		79	
64	310		538	
65	336		335	
66	305		337	
67	143		380	
68	195		369	377
69	175		371	
70	174		372	
71	142		315	

FMT

[illegible]

961	400	PMI	407		
962	406	PMI	405		
973	346	PMI	147	417	511
990	6	PMI	392		
1000	364	PMI	5		
1001	400	PMI	363		
1002	525	PMI	409		
			526		

## \*\*\* TABUL DICTIONARY -- TRANS \*\*\*

JAPN	TYFM	TYFF	REFERENCES,
100	14		20
200	9		7